MIXED MODELS, POSTERIOR MEANS AND PENALIZED LEAST SQUARES

A Dissertation

by

YOLANDA MUÑOZ MALDONADO

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2005

Major Subject: Statistics

MIXED MODELS, POSTERIOR MEANS AND PENALIZED LEAST SQUARES

A Dissertation

by

YOLANDA MUÑOZ MALDONADO

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Commitee,	Randall L. Eubank
Committee Members,	Raymond J. Carroll
	Michael Sherman
	Suojin Wang
	Joseph D. Ward
Head of Department,	Simon J. Sheather

August 2005

Major Subject: Statistics

ABSTRACT

Mixed Models, Posterior Means and Penalized Least Squares. (August 2005)
Yolanda Muñoz Maldonado, B.S., Universidad Autonoma de Yucatan;
M.S., The University of Texas at El Paso
Chair of Advisory Committee: Dr. Randall L. Eubank

In recent years there has been increased research activity in the area of Functional Data Analysis. Methodology from finite dimensional multivariate analysis has been extended to the functional data setting giving birth to Functional ANOVA, Functional Principal Components Analysis, etc. In particular, some studies have proposed inferential techniques for various functional models that have connections to well known areas such as mixed-effects models or spline smoothing. The methodology used in these cases is computationally intensive since it involves the estimation of coefficients in linear models, adaptive selection of smoothing parameters, estimation of variances components, etc.

This dissertation proposes a wide-ranging modeling framework that includes many functional linear models as special cases. Three widely used tools are considered: mixed-effects models, penalized least squares, and Bayesian prediction. We show that, in certain important cases, the same numerical answer is obtained for these seemingly different techniques. In addition, under certain assumptions, an application of a Kalman filter algorithm is shown to improve the order of computations, by two orders of magnitude, for point and interval estimates (with n being the sample size). A functional data analysis setting is used to exemplify our results.

To my parents, Ignacio and Yolanda.

ACKNOWLEDGEMENTS

During the years that took me to obtain this degree, there were some times I felt overwhelmed by different situations: my dad's stroke, financial problems, the demanding working hours. If it were not for all the people that surrounded me, I would not had been able to finish. This is just a small way of saying thanks to all of you who helped me on this journey.

To Tina Schuster, who drove with me from El Paso, and made sure that I was safely settled in A&M. To all the staff in the Statistics Department, specially Marilyn, Jennifer, Elaine and Sandra. I know that their handling of administrative problems made my life here easy. To all my professors: Dr. Longnecker, Dr. Hart, Dr. Hardin, Dr. Cline, Dr. Mallick, Dr. Wehrly, Dr. Newton, Dr. Wang, Dr. Hsing, Dr. Claeskens, Dr. Vanucci, Dr. Calvin, Dr. Sherman, and especially Dr. Spiegelman for making me realize that I could give a greater effort than the one I thought I was capable of.

Thanks to all my committee members for all their suggestions and patience. To my friends Fernando, Brisa, and Paty for their moral support. To Alex, whose unlimited patience, understanding and love helped me to go through the hard times. To my family, specially my parents, who always told me that I could attain whatever I set my mind and heart on.

My last thank you goes to my advisor and friend, Randy Eubank. There are no words to describe the magnitude of his support. I can just say that it was his help, advise and friendship the key factors that allowed me to accomplish my dream.

Thanks to all of you because I could not have done it without your help!

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

In recent years there has been increased research activity in the area of Functional Data Analysis (FDA). Methodology from finite dimensional multivariate analysis has been extended to the infinite dimensional functional data setting giving birth to Functional ANOVA, Functional Principal Components Analysis, etc. With the development of this new methodology two issues have emerged: 1) the need to integrate the theory developed so far into a general framework and 2) implementation of efficient algorithms to compute corresponding point and interval estimators.

Different classes of proposed functional models are associated with well known inferential methods derived from mixed-effects models or spline smoothing. In this dissertation we focus on three powerful statistical tools for inference in what are seemingly very different settings: mixed-effects models, smoothing spline estimation and prediction in a particular Gaussian signal-plus-noise model having a parametric linear trend modeled with a diffuse prior (we will refer to this model as the Bayesian model for the remainder of this dissertation). We use these three tools to build a more efficient and general framework that can be applied in several model scenarios.

1.1 Mixed-effects Model

Mixed-effects models are widely used in applied Statistics and are the more general case of analysis of variance models in that they combine the fixed and the random

---

This dissertation follows the style and format of *Biometrics*.

effects models. During the second decade of the 20th century, Fisher developed the basic principles used in the Analysis of Variance (ANOVA). In this setting we assume that observed responses from experimental units can be written as a linear combination of some unknown parameters. The parameters being estimated in the model can be considered as constants or as realizations derived from a random process.

The linear model approach has been applied to a variety of problems: assessing the variability in a data set according to some levels or treatments of another variable which may or may be not random, using repeated measures or longitudinal studies, etc. Books such as *"The Theory and Application of the Linear Model"* (Graybill, 1976), *"The Analysis of Variance"* (Scheffé, 1959), *"Methods and Applications of Linear Models"* (Hocking, 1996) and more recently, *"Mixed Models"* (Demidenko, 2004), present an ample discussion of the methodology used in the analysis of linear models.

The following are examples taken from Hocking's (1996) book illustrating the application of the linear model in different settings.

1. **Two Factor Mixed Model**. Sheffé (1959) considered the production of a factory involving different machines and different operators. He considered a setting where he had A different machines and B operators. The actual machines were fixed but the operator influence was considered as random. Each operator can run a machine K times. The linear model formulation is then given by:

$$y_{ijk} = \alpha_i + b_j + (ab)_{ij} + e_{ijk}, \tag{1.1}$$

where α_i represents the fixed effect of the i^{th} machine, b_j is the random effect corresponding to the j^{th} operator, $(ab)_{ij}$ is the random interaction between machine i and operator j, and the e_{ijk} represent independent errors with zero mean and common variance σ_e^2. The random effects for operators and interactions are

assumed to both have zero mean and respective variances σ_b^2 and $\sigma_{(ij)}^2$.

2. **Repeated Measures**. For this example we consider the effect of three different drugs on patients with a heart condition. Patients are randomly assigned to each drug and, after taking the drug, the patient's heart beat is measured four times at five minutes intervals. We are interested in the effect of the drugs and their behavior over time.

   We can write a linear model for this problem as:

$$y_{ijk} = \alpha_i + b_{j(i)} + e_{ijk}, \tag{1.2}$$

   with $i = 1, 2$, $j = 1, \ldots, n_i$, and $k = 1, \ldots, 4$. The i^{th} drug factor is represented by α_i, $b_{j(i)}$ represents the effect of patient j assigned to drug i and the e_{ijk} are independent random errors. We have that

$$\text{Var}[y_{ijk}] = \sigma_{b_j}^2 + \sigma_e^2,$$

   and

$$\text{Cov}[y_{ijk}, y_{i'j'k'}] = \begin{cases} \sigma_{jj'}, & \text{if} \quad i = i' \text{ and } j = j', \\ 0, & \text{if} \quad i \neq i' \text{ or } j \neq j'. \end{cases}$$

Regardless of the context in which we use the mixed-effects model, it is always possible to write the model in a general form using matrices. For this purpose, let $\boldsymbol{y}$ be a $n \times 1$ vector of responses and let n be the total number of observations. Denote by T the design matrix for the fixed effects and take U to be the design matrix for the random effects. Let $\boldsymbol{\theta}$ be a $m \times 1$ vector of unknown parameters that represents the fixed effects means, use $\boldsymbol{\gamma}$ to denote a $q \times 1$ vector of random effects and let $\boldsymbol{e}$ be a $n \times 1$ vector of random errors. The mixed-effects model is then,

$$\boldsymbol{y} = T\boldsymbol{\theta} + U\boldsymbol{\gamma} + \boldsymbol{e}. \tag{1.3}$$

Here the random effects are assumed to have moments given by:

$$E[\boldsymbol{\gamma}] = \mathbf{0}, \tag{1.4}$$

$$E[\boldsymbol{\gamma}\boldsymbol{\gamma}^T] = \begin{bmatrix} \sigma_1^2 R_1 & \dots & \dots & 0 \\ 0 & \sigma_2^2 R_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_b^2 R_b \end{bmatrix}, \tag{1.5}$$

where R_j, $j = 1, \dots, b$, are known symmetric matrices and the $\sigma_j^2 > 0$ are the variance components for the random effects.

To simplify the notation, in much of the sequel we will assume that there is only one random effect. In that case, we will set $\sigma_1^2 = \sigma_b^2$ so that

$$E[\boldsymbol{\gamma}\boldsymbol{\gamma}^T] = \sigma_b^2 R. \tag{1.6}$$

The random errors then have

$$E[\boldsymbol{e}] = \mathbf{0}, \tag{1.7}$$

and

$$E[\boldsymbol{e}\boldsymbol{e}^T] = \sigma_e^2 I, \tag{1.8}$$

with I the identity matrix and $\sigma_e^2 > 0$ the error variance component. As a result, the first two moments of $\boldsymbol{y}$ are given by

$$E[\boldsymbol{y}] = T\boldsymbol{\theta}, \tag{1.9}$$

$$Var[\boldsymbol{y}] = \sigma_b^2 URU^T + \sigma_e^2 I. \tag{1.10}$$

This representation encompasses more complicated models, like nested models (our second example) or models with interaction between the fixed and random effects

(example 1). But basically all of them can be written in the form of (1.3). Consider our example 1 and let us assume that we have two machines, with two operators and two repetitions for each machine-operator case. Then we have

$$\boldsymbol{y} = \begin{bmatrix} y_{111} \\ y_{112} \\ y_{121} \\ y_{122} \\ y_{211} \\ y_{212} \\ y_{221} \\ y_{222} \end{bmatrix}, \quad \boldsymbol{e} = \begin{bmatrix} e_{111} \\ e_{112} \\ e_{121} \\ e_{122} \\ e_{211} \\ e_{212} \\ e_{221} \\ e_{222} \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} \quad \text{and } \boldsymbol{\gamma} = \begin{bmatrix} b_1 \\ b_2 \\ ab_{11} \\ ab_{12} \\ ab_{21} \\ ab_{22} \end{bmatrix}$$

with

$$T = \begin{bmatrix} \mathbf{1}_{1\times 4} & \mathbf{0}_{1\times 4} \\ \mathbf{0}_{1\times 4} & \mathbf{1}_{1\times 4} \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} \mathbf{1}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{1}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} \\ \mathbf{0}_{1\times 2} & \mathbf{1}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{1}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} \\ \mathbf{1}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{1}_{1\times 2} & \mathbf{0}_{1\times 2} \\ \mathbf{0}_{1\times 2} & \mathbf{1}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{1}_{1\times 2} \end{bmatrix}, \quad (1.11)$$

where $\mathbf{0}_{1\times n}$ is the vector with n elements equal to zero, and $\mathbf{1}_{1\times n}$ is the vector with all n elements equal to unity.

The variance-covariance matrix of the random errors is given by $\sigma_e^2 I_8$ and the

variance-covariance matrix of the random effects is

$$R = \begin{bmatrix} \sigma^2_{b_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{b_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{21} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{22} \end{bmatrix}. \tag{1.12}$$

For our second example we have a design with two treatments, three patients per treatment and 4 measurements per patient so our matrices are of the form

$$\boldsymbol{y} = \begin{bmatrix} y_{111} \\ \vdots \\ y_{114} \\ y_{121} \\ \vdots \\ y_{134} \\ \vdots \\ y_{211} \\ \vdots \\ y_{234} \end{bmatrix}, \quad \boldsymbol{e} = \begin{bmatrix} e_{111} \\ \vdots \\ e_{114} \\ e_{121} \\ \vdots \\ e_{134} \\ \vdots e_{211} \\ \vdots \\ e_{234} \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}, \quad \text{and } \boldsymbol{\gamma} = \begin{bmatrix} b_{1(1)} \\ b_{2(1)} \\ b_{3(1)} \\ b_{1(2)} \\ b_{2(3)} \\ b_{3(2)} \end{bmatrix}$$

with

$$T = \begin{bmatrix} \mathbf{1}_{1\times 4} & \mathbf{0}_{1\times 4} \\ \mathbf{0}_{1\times 4} & \mathbf{1}_{1\times 4} \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} \mathbf{1}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} \\ \mathbf{0}_{1\times 2} & \mathbf{1}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} \\ \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{1}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} \\ \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{1}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} \\ \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{1}_{1\times 2} & \mathbf{0}_{1\times 2} \\ \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{0}_{1\times 2} & \mathbf{1}_{1\times 2} \end{bmatrix}.$$

The variance-covariance matrix of the random errors is given by $\sigma_e^2 I_{24}$ and the variance-covariance matrix of the random effects is as in (1.5) with $i = 1, \ldots, 6$ and

$$R_i = \begin{Bmatrix} 1 & \sigma_{ii} \\ \sigma_{ii} & 1 \end{Bmatrix}.$$

Generally, it is of interest to estimate the vector of fixed effects, $\boldsymbol{\theta}$, the variance components and to predict the random effects or estimable linear combinations of both fixed and random effects. Popular inferential methods for these type of models are Maximum Likelihood Estimation (MLE), Restricted Maximum Likelihood Estimation (REML) and Least Squares Estimation (LSE). In the remainder of this section we will illustrate how these techniques are applied to the fixed-effects model, the random-effects model and, finally, to the more general case of the mixed-effects model.

To analyze the data under the assumptions of a linear fixed-effects or ANOVA model we make use of the methods of Maximum Likelihood (ML) or Least Squares (LS) to obtain estimators for the unknown fixed parameters and for the single variance component. A fixed-effects model can be written as

$$\boldsymbol{y} = T\boldsymbol{\theta} + \boldsymbol{e}, \tag{1.13}$$

with the moments of $\boldsymbol{e}$ as in (1.7) and (1.8). The MLE (under normality assumptions) and LSE yield the same solution for estimation of $\boldsymbol{\theta}$: i.e., the estimator is

$$\hat{\boldsymbol{\theta}} \;=\; (T^T T)^{-1} T^T \boldsymbol{y}.$$

The Gauss-Markov theorem states that this is also the best among all unbiased linear estimators (BLUE) of the fixed parameters in the sense of having minimum variance.

When T has less than full rank it means that some of the columns of T can be written as linear combinations of the other columns. In this case, we can remove the columns that are not linearly independent, impose some restriction on the unknown parameters or use a generalized inverse of $T^T T$. The estimator $\hat{\boldsymbol{\theta}}$ is not going to be unique, but estimators of the estimable linear combinations of $\boldsymbol{\theta}$ will be invariant.

Estimation of the single variance component can be done by maximizing the likelihood of $\boldsymbol{y}$ with respect to σ_e^2 (and plugging in the value of $\hat{\boldsymbol{\theta}}$ instead of the parameter) or using least squares. In the LS setting, we estimate σ_e^2 using the LS estimator of $\boldsymbol{\theta}$ and adjusting it for bias: i.e., we estimate σ_e^2 by

$$s^2 \;=\; \frac{(\boldsymbol{y} - T\hat{\boldsymbol{\theta}})^T (\boldsymbol{y} - T\hat{\boldsymbol{\theta}})}{n - m}. \tag{1.14}$$

The difference between the s^2 and the MLE estimator for σ_e^2 is that the MLE is biased. But, since it is a complete sufficient statistic, we can make it unbiased.

Using our estimator for σ_e^2 along with the distributional assumptions on the random errors, allows us to obtain confidence intervals or regional estimates such that we can have an idea of plausible values for the parameter. In this respect, there are two possible approaches we can take for interval estimation:

1. For each value of θ_i, $i = 1, \ldots, m$, we can find $(1-\alpha)100\%$ confidence intervals and treat these intervals separately.

2. We want simultaneous coverage for all values of θ_i so that the m corresponding intervals will have a probability of $(1-\alpha)$ of covering all the parameters at the same time.

For the first case, what we usually do is to obtain separate t confidence intervals for each θ_i. In the simultaneous case there exists several methods which depend on if we are interested just in a particular set of confidence regions, in which case we will apply Bonferroni's confidence intervals for example, or, if we are interested in confidence regions encompassing all possible set of contrasts. In this latter situation we can use Scheffé's method. There exist abundant literature on this topic and we refer the reader to Hocking (1996) for a more extensive discussion.

From this brief review we can see that in the fixed-effects model our main interest is to make inference about the mean of $\boldsymbol{y}$. Under the random-effects model our interest lies in the variance components. In the random-effects model we have a vector of observations $\boldsymbol{y}$ such that

$$\boldsymbol{y} \;=\; U\boldsymbol{\gamma} + \boldsymbol{e}, \tag{1.15}$$

with the moments of $\boldsymbol{e}$ and $\boldsymbol{\gamma}$ as in (1.7), (1.8), (1.4) and (1.6), respectively. Often times, the variance components are parameterized as:

$$\lambda_1 \;=\; \sigma_e^2, \tag{1.16}$$

and

$$\lambda_2 \;=\; \frac{\sigma_b^2}{\sigma_e^2}. \tag{1.17}$$

Under these model assumptions, the problem of interest is to predict $\boldsymbol{\gamma}$ and to estimate the variance components σ_b^2 and σ_e^2 or, equivalently, λ_1 and λ_2.

The early works on estimation of variance components, like Eisenhart (1947),

Henderson (1959) and Tukey (1956), usually estimated the unknown variances by computing the mean squares and equating them to their expectations. A downside of this method is the possibility of obtaining negative component estimators.

Harville, in 1977, discussed the ML approach to variance component estimation. He pointed out that application of the ML method allows one to incorporate the non-negativity constraints in the parameter space without difficulty and the MLE's can be easily obtained for any given parameterizations of the model. He also mentioned that one of the reasons the ML method has not been broadly used was due to the computational effort in obtaining the MLE's (since they are a numerical solution to a nonlinear optimization problem with constraints).

Real problems gave birth to the combination of fixed and random effects. Henderson (1953) talks about three methods for estimating variance components in the mixed-effects setting:

- **Method I**: Estimate fixed and random effects as in the usual ANOVA procedure, i.e., considering the random effects as fixed.
- **Method II**: Estimate the fixed effects via LS and then apply method I to the modified data.
- **Method III** Similar to method I but instead of using standard ANOVA methods, use methods for non-orthogonal data like the method of fitting constants or weighted squares of means.

Method I and method II yield biased estimators whereas method III produces unbiased estimators at the cost of lengthy computations.

Later, Henderson (1959) and Hartley and Rao (1967) proposed the method of maximum likelihood for estimating the parameters of interest. But the ML estimators

were still biased because they do not take into account the degrees of freedom lost in the estimation of the fixed effects.

Patterson and Thompson (1971), suggested a modification of the method that gives unbiased estimators. Instead of using the likelihood of $\boldsymbol{y}$, the REML method considers the likelihood of a particular linear transformation of $\boldsymbol{y}$. The later method would be named Restricted Maximum Likelihood (REML) by Corbeil and Searle (1976).

As with the computation of the MLE's, computation of REML estimators are based on iterative numerical procedures. There is no "best" algorithm. Sometimes, an algorithm that converges fast for one case, will fail to converge in another. There are cases when the computational effort is too demanding so that it is better to use some type of approximation approach to the REML as proposed by Harville (1977).

From the discussion above, it can be seen that one of the main concerns is the computations involved in obtaining variance component estimators. Although advances in technology have made it possible to address such complex computing problems, the same technological advances allow us to gather much larger data sets than before. So it is still an issue to find computationally efficient ways of obtaining those estimators.

1.2 Spline Smoothing

Smoothing spline estimation has been widely used in nonparametric regression. These estimators arose as a numerical analysis tool and got some attention with the work of Schoenberg (1964). But, it was not until the work of Wahba, in the early 1970's, that smoothing splines caught the attention of statisticians and, from that point, research in the area has been prolific. See, e.g., Wahba (1978, 1983, 1985, 1990), Eubank (1988, 1996), Speckman (1985) and Silverman (1985) for examples of such work and

references.

Smoothing splines are function estimators derived from a penalized least squares error criterion. These estimators provide a way of balancing a good approximation to the data and a certain degree of smoothness in the fitted curve. To be more specific, first let $y(t_1), y(t_2), \ldots, y(t_n)$ be responses of a stochastic process observed at ordinates $0 \leq t_1 < \cdots < t_n \leq 1$. It is then common to use a signal-plus-noise representation for the response: i.e.

$$y(t_i) \quad = \quad f(t_i) + e(t_i), \tag{1.18}$$

for $i = 1, \ldots, n$, where the $e(t_i)$'s are zero mean uncorrelated random variables with common variance σ_e^2. It is often assumed that $f(\cdot)$ is a member of the set of all continuously differentiable functions with square integrable second derivatives.

Take $\boldsymbol{y} = [y(t_1), \ldots, y(t_n)]^T$, $\boldsymbol{f} = [f(t_i), \ldots, f(t_n)]^T$. Then, the smoothing spline criterion for estimating a function g with a square integrable second derivative is

$$L(\boldsymbol{g}) \quad = \quad (\boldsymbol{y} - \boldsymbol{g})^T(\boldsymbol{y} - \boldsymbol{g}) + \frac{1}{n\lambda}\int_0^1 [g^{(2)}(t)]^2 dt, \tag{1.19}$$

for $\boldsymbol{g} = [g(t_1), \ldots, g(t_n)]$. This criterion, $L(\boldsymbol{g})$, is minimized over all functions g belonging to the space of functions having square integrable second derivatives to obtain the estimator $\hat{\boldsymbol{f}}$ of $\boldsymbol{f}$. The value of $\lambda \geq 0$ in (1.19) governs the trade off between the fit of the estimator and the smoothness of the fitted function.

The minimizer of (1.19) is well known to be a natural cubic spline when the criterion is minimized over all functions with two continuous derivatives. We can represent the natural cubic spline as

$$s(t) = \sum_{j=1}^{2} \theta_j \phi_j(\cdot) + \sum_{i=1}^{n} b_i \xi_i(\cdot),$$

where θ_j, $\phi_j(\cdot)$, b_i, and $\xi_i(\cdot)$ are defined explicitly in Chapter II. For now, it will suffice to know that $\phi_j(\cdot)$ and $\xi_i(\cdot)$ form together a basis for the space of polynomial

splines of order 4 with knots at $t_1, \ldots, t_n$ subjected to the restrictions

$$s^{(j)}(0) = s^{(j)}(1) = 0$$

for $j = 2, 3$. The components θ_j and b_i are coefficients for their respective basis functions.

Now, let

$$T = \{\phi_j(t_i)\}_{\substack{i=1,n \\ j=1,2}},$$

and

$$R = \{\xi_i(t_j)\}_{i,j=1,n}.$$

The smoothing spline estimator of $\boldsymbol{f}$ is then given by

$$\hat{\boldsymbol{f}} = [I - Q^{-1}(I - T(T^T Q^{-1} T)^{-1} T^T Q^{-1})]\boldsymbol{y}, \tag{1.20}$$

or its alternative representation

$$\hat{\boldsymbol{f}} = B(B^T Q B)^{-1} B^T \boldsymbol{y}, \tag{1.21}$$

where $Q = (n\lambda R + I)$ and B is any $n \times (n-2)$ matrix of rank $(n-2)$ satisfying

$$B^T T = 0.$$

One criticism often heard about smoothing splines is that, since smoothing splines take as knots the design points t_i, the algorithms to obtain $\hat{\boldsymbol{f}}$ are computationally demanding. In theory, it has always being possible to use splines of order higher than 4, but the computational aspects were a big factor to consider, making it a custom to use only cubic smoothing splines. To ease this burden, people like Kimeldorf and Wahba (1970), Anselone and Laurent (1968), Reinsch (1967) and Greville

(1969) studied different type of spline basis functions.

Different choices of B in (1.21) will yield diverse types of basis functions (Eubank, 1988, 1996). For example, Kimeldorf and Wahba proposed to use basis functions choosing B such that $B^TB = I$ (with $B^TT = 0$ as before), to obtain more stable numerical computations. The basis functions are then given by $1, t$ and $g_1(t), \ldots, g_{(n-2)}(t)$ where

$$[g_1(t), \ldots, g_{(n-2)}(t)]^T \quad = \quad QB.$$

Anselone, Laurent and Reinsch chose a matrix B such that $B(B^TQB)^{-1}B^T$ is 5 banded. The ith row of B is of the form

$$\boldsymbol{B}_i \quad = \quad (0, \ldots, \alpha_{0,2}[i], \alpha_{1,2}[i], \alpha_{2,2}[i], 0, \ldots, 0),$$

where $\alpha_{k,2}[i]$, $k = 0, 1, 2$, is the nonzero 2nd order normalized divided difference coefficients of f at t_i. The banded structure in Q that is produced by this choice for B can be used to obtain efficient computations. The Anselone, Laurent and Reinsch basis has what is called a local support property that makes them well suited for numerical applications (Schumaker, 1981).

These approaches are just examples of what people tried to do to ease the computational burden of obtaining the smoothing spline estimator. However, computation of the smoothing spline estimator is not the only task on hand. Before obtaining the smoothing spline estimator, we need to select the value of the smoothing parameter and, since it involves quantities used for calculating $\hat{\boldsymbol{f}}$, it is also computationally taxing.

Automatic, data-driven methods for selecting the value of λ include Generalized Cross Validation (GCV), Generalized Maximum Likelihood (GML) and Unbiased Risk Prediction (UBR). These methods permit one to adaptively choose the levels of

smoothing based on the data and, hence, as the number of data points gets larger the required amount of computational effort will also grow.

As in the case of the estimation for mixed-effects model parameters, the improvements in computational technology have alleviated, to some degree, the computational burden of obtaining the smoothing spline estimators. But, when FDA started to gain popularity in the 1980's, computational issues once again rose to the forefront. Now, instead of dealing with one curve to estimate, we were dealing with many, many curves. So the problem of finding ways to efficiently compute the estimator for the curves and select their respective smoothing parameters has again become important.

1.3 Bayesian Model

Wahba (1978) found that the fitted curve obtained in a non-parametric regression setting using smoothing splines is numerically identical to the posterior mean of an integrated Brownian motion signal plus a polynomial drift modeled using a diffuse prior. More precisely, consider the model

$$y(t_i) \;=\; \sum_{j=1}^{2} \theta_j \phi_j(t_i) + \sigma_b X(t_i) + e(t_i) \tag{1.22}$$

with $X(t)$, $t \in [0,1]$, a zero-mean Gaussian stochastic process with covariance function

$$\mathrm{E}[X(t_i)X(t_j)] \;=\; \int_0^{\min(t_j,t_i)} \frac{(t_j-u)(t_i-u)}{[(2-1)!]^2} du. \tag{1.23}$$

The functions $\phi_j(\cdot)$, are polynomial terms of the form: $\phi(t) = t^{j-1}/(j-1)!$, $j = 1, 2$; the coefficients θ_j are modeled as uncorrelated normal random variables with zero mean and variance ν. The $e(t_i)$'s are uncorrelated normally distributed random errors with zero mean and variance σ_e^2 that are independent of $X(\cdot)$ and the θ_j's.

Writing model (1.22) in matrix form we get

$$\boldsymbol{y} \;=\; T\boldsymbol{\theta} + \sigma_b \boldsymbol{X} + \boldsymbol{e}, \tag{1.24}$$

where $\boldsymbol{y}$ and $\boldsymbol{e}$ are as in (1.3) and $\boldsymbol{X} = [X(t_1), \ldots, X(t_n)]^T$. Here

$$T \;=\; \{\phi_j(ti)\}_{\substack{i=1,n \\ j=1,2}},$$

and the variances of $\boldsymbol{X}$ and $\boldsymbol{e}$ are given by

$$\mathrm{Var}(\sigma_b \boldsymbol{X}) \;=\; \sigma_b^2 R$$

and

$$\mathrm{Var}(\boldsymbol{e}) \;=\; \sigma_e^2 I,$$

respectively. The matrix R has elements of the form $\{\mathrm{E}[X(t_i)X(t_j)]\}_{i,j=1,n}$.

Notice that model (1.24) is now a complete random-effects model of the form (1.3) with

$$U = [T, I] \quad \text{and} \quad \boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{\theta} \\ \sigma_b^2 \boldsymbol{X} \end{bmatrix}.$$

Consequently, we have three variance components to estimate, i.e.: σ_e^2, σ_b^2 and ν.

From the Bayesian perspective, we are imposing a *prior* distribution for $\boldsymbol{\theta}$. Generally speaking, there are two ways we can select for specifying the distribution of $\boldsymbol{\theta}$. First, we can choose a proper prior on $\boldsymbol{\theta}$, in which case we can work the problem in the same way we treat a mixed-effects model (i.e, using the method of REML or LS) or, we can try to use the method of GML, which uses the conditional distributions of the random variables. In fact, we will show in Chapter II that the REML method and the GML method give equivalent results when obtaining the variance components of (1.24).

Instead of assuming a proper prior on $\boldsymbol{\theta}$, on can proceed as in Wahba (1978) and assume a diffuse prior for $\boldsymbol{\theta}$. This can be accomplished by letting $\nu \to \infty$. The posterior mean of $\sum_{j=1}^{2} \theta_j \phi_j(t_i) + \sigma_b X(t_i)$, in this case, does not involve ν and it

becomes numerically equivalent to (1.20) (we will show this fact in Chapter II).

The estimation of the variance components in this Bayesian setting turns out to be equivalent to the estimation of the variance of the random errors σ_e^2 and the smoothing parameter. This can be done via GML in combination with the prior information on the θ_j's and parameterizing the likelihood in terms of σ_e^2 and $\lambda = \sigma_b^2/\sigma_e^2$. More explicitly, using the prior distribution of $\boldsymbol{\theta}$ we can obtain the likelihood of $\boldsymbol{y}$ and then find a suitable matrix P such that the likelihood of $P\boldsymbol{y}$ can be partitioned into two components: one involving $\boldsymbol{\theta}$ and the variances components and the other involving only σ_b^2 and σ_e^2. We then parameterize the likelihood in terms of σ_e^2 and λ and consider the later fixed. Then the likelihood is maximized with respect to σ_e^2. This estimator is then plugged into the likelihood, which now is maximized with respect to λ. In this way, we can find estimators of the variance components with the portion of the likelihood that does not involve the diffuse part of the model.

There are very interesting connections between the Bayesian model and the estimator obtained by LS and the mixed-effects model. Kimeldorf and Wahba (1970) showed that, when we consider (1.22) as a function of t, the Best Linear Unbiased Predictor (BLUP) of $y(t)$ is also the minimizer of the criterion (1.19) with smoothing parameter equal to the ratio of the variance components σ_b^2 and σ_e^2. Robinson (1991) made another interesting connection between the Bayesian model and model (1.3). He noticed that the joint density of $\boldsymbol{y}$ and $\boldsymbol{\gamma}$ in the mixed-effects model (1.3) is proportional to the likelihood of $\boldsymbol{y}$ given $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ in model (1.24) when $\boldsymbol{\theta}$ has a non-informative prior distribution . This will indicate that the BLUP estimates of (1.3) are numerically equivalent, not only to the smoothing spline estimator, but to the posterior mean of (1.24).

1.4 Dissertation Goals

We are proposing a general framework for the theory developed so far in FDA. However, this framework is also general enough that it can be applied in the mixed-effects model, penalized least squares or Bayesian model settings. Once we establish this general framework, we show that, if we are willing to assume a certain structure in our model, we can use the Kalman filter to obtain the desired estimators and their respective variances all in order n operations. The use of these efficient algorithms permit us to efficiently compute the likelihood of the responses and the diagonal elements of the "hat matrix" in the LS technique. These quantities are used via GML or GCV, respectively, to obtain data driven choices for the order of spline and the level of smoothing when estimating functional models, all in $O(n)$ calculations.

In Chapter II we show the connection between the mixed-effects model, the smoothing spline estimators and the Bayesian model. We also illustrate the equivalence between the GML and the REML methods for obtaining variance components. In Chapter III we state a theorem that extends this connection to a general mixed-effects model setting. This theorem will allow us to broaden existing methodology to the penalized least squares error criterion, functional models with correlated random errors and varying coefficient models. We also discuss the application of the GML or the GCV criteria to obatining estimators for variance components and/or smoothing parameters in the different settings. Finally, in Chapter IV, we introduce the concept of state-space structure and show efficient Kalman filter algorithms. We provide detailed descriptions for a number of particular model scenarios and illustrate how to use the Kalman filter recursions implemented in SAS and R software.

CHAPTER II

BACKGROUND

Since the early development of some of the theoretical results for smoothing splines (e.g., Wahba, 1978) it has been known that there is an intimate connection between estimators resulting from spline smoothing and those using a particular Gaussian signal-plus-noise model having a polynomial trend in which an improper prior is assumed. The work of Harville (1976) explains that in certain cases, when using a mixed effects model, it is advisable to put a diffuse prior on the fixed effects and Speed (1991) pointed out the relationship between REML estimators and BLUPs.

In this section we show the connection between the three tools mentioned in Chapter I and we will establish the relationship between smoothing parameter selection for a smoothing spline in non-parametric regression, the method of REML in a mixed effects model, with a specific covariance structure and normality assumptions, and the GML method applied to the Bayesian model for estimation of the variance components. Specifically, we will show that smoothing parameter selection by GML is tantamount to estimation of the variance components using REML.

2.1 Mixed Models Approach

In this section we will briefly review the procedure to estimate the fixed and random components in a mixed-effects model with certain assumptions. Consider the mixed-effects model

$$\boldsymbol{y} \;=\; T\boldsymbol{\theta} + \boldsymbol{\gamma} + \boldsymbol{e}, \tag{2.1}$$

where $\boldsymbol{y}$ is the $n \times 1$ vector of responses, T is the design matrix for the fixed effects of dimension $n \times m$, $\boldsymbol{\theta}$ is an $m \times 1$ vector of fixed effects, $\boldsymbol{\gamma}$ is a $n \times 1$ vector of random effects and $\boldsymbol{e}$ is an $n \times 1$ vector of random errors which are normally distributed with zero mean and variance $\sigma_e^2 I$. Also, $\boldsymbol{\gamma}$ is normally distributed with zero mean and variance $\sigma_b^2 R$ and it is uncorrelated with $\boldsymbol{e}$.

Thus, the moments of $\boldsymbol{y}$ are found to be

$$\mathrm{E}(\boldsymbol{y}) \;=\; T\boldsymbol{\theta}, \tag{2.2}$$

and

$$\mathrm{Var}(\boldsymbol{y}) \;=\; \sigma_b^2 R + \sigma_e^2 I. \tag{2.3}$$

Now, rewrite $\mathrm{Var}(\boldsymbol{y})$ as

$$\mathrm{Var}(\boldsymbol{y}) \;=\; \sigma_e^2 \left(I + n\lambda R\right), \tag{2.4}$$

$$\;=\; \sigma_e^2 Q, \tag{2.5}$$

for

$$n\lambda \;=\; \frac{\sigma_b^2}{\sigma_e^2}, \tag{2.6}$$

and

$$Q \;=\; n\lambda R + I. \tag{2.7}$$

The density of $\boldsymbol{y}$ is then given by

$$\mathrm{L}(\boldsymbol{y}) \;=\; (2\pi)^{-n/2}(\sigma_e^2)^{-1/2}|Q|^{-1/2}\exp\left\{-\frac{1}{2\sigma_e^2}(\boldsymbol{y}-T\boldsymbol{\theta})^T Q^{-1}(\boldsymbol{y}-T\boldsymbol{\theta})\right\}. \tag{2.8}$$

Let us first consider σ_e^2 and λ to be fixed and we will estimate them via REML subsequently. Using the method of LSE we obtain normal equations for the fixed-effects of the form

$$-T^T Q^{-1}(\boldsymbol{y}-T\boldsymbol{\theta}) \;=\; \mathbf{0}, \tag{2.9}$$

from which we obtain the Best Linear Unbiased Estimator (BLUE) of $\boldsymbol{\theta}$

$$\hat{\boldsymbol{\theta}} = (T^T Q^{-1} T)^{-1} T^T Q^{-1} \boldsymbol{y}. \tag{2.10}$$

To find a predictor for $\boldsymbol{\gamma}$ we need to find the conditional distribution of $\boldsymbol{\gamma}$ given $\boldsymbol{y}$. We already know the density of $\boldsymbol{y}$. The probability density function for $\boldsymbol{\gamma}$ is

$$\mathrm{L}(\boldsymbol{\gamma}) = (2\pi)^{-n/2}(\sigma_e^2)^{-n/2}|n\lambda R|^{-1/2}\exp\left\{-\frac{1}{2\sigma_e^2}\boldsymbol{\gamma}^T(n\lambda R)^{-1}\boldsymbol{\gamma}\right\}, \tag{2.11}$$

so that the density of $\boldsymbol{y}$ given $\boldsymbol{\gamma}$ is

$$\mathrm{L}(\boldsymbol{y}|\boldsymbol{\gamma}) = (2\pi)^{-n/2}(\sigma_e^2)^{-n/2}\exp\left\{-\frac{1}{2\sigma_e^2}(\boldsymbol{y}-T\boldsymbol{\theta}-\boldsymbol{\gamma})^T(\boldsymbol{y}-T\boldsymbol{\theta}-\boldsymbol{\gamma})\right\}. \tag{2.12}$$

The joint distribution of $\boldsymbol{y}$ and $\boldsymbol{\gamma}$ is then seen to be equal to

$$\begin{aligned}\mathrm{L}(\boldsymbol{y},\boldsymbol{\gamma}) &= |n\lambda R|^{1/2} \\ &\quad \times\exp\left\{-\frac{1}{2\sigma_e^2}(\boldsymbol{y}-T\boldsymbol{\theta}-\boldsymbol{\gamma})^T(\boldsymbol{y}-T\boldsymbol{\theta}-\boldsymbol{\gamma}) - \frac{1}{2\sigma_e^2}\boldsymbol{\gamma}^T(n\lambda R)^{-1}\boldsymbol{\gamma}\right\}.\end{aligned} \tag{2.13}$$

Using this density and the density of $\boldsymbol{y}$ we get the conditional density for $\boldsymbol{\gamma}$ given $\boldsymbol{y}$ to be

$$\begin{aligned}\mathrm{L}(\boldsymbol{\gamma}|\boldsymbol{y}) \propto\ & \exp\left\{-\frac{1}{2\sigma_e^2}(\boldsymbol{y}-T\boldsymbol{\theta}-\boldsymbol{\gamma})^T(\boldsymbol{y}-T\boldsymbol{\theta}-\boldsymbol{\gamma}) - \frac{1}{2\sigma_e^2}\boldsymbol{\gamma}^T(n\lambda R)^{-1}\boldsymbol{\gamma}\right. \\ & \left. +\frac{1}{2\sigma_e^2}(\boldsymbol{y}-T\boldsymbol{\theta})^T Q^{-1}(\boldsymbol{y}-T\boldsymbol{\theta})\right\},\end{aligned} \tag{2.14}$$

where $\propto$ stands for "proportional to".

Using the Sherman-Morrison-Woodsbury formula (Householder, 1964, pp.124)

$$(A+BC^{-1}D)^{-1} = A^{-1} - A^{-1}B(C+DA^{-1}B)^{-1}DA^{-1}$$

and letting $A = I$, $B = I$, $C^{-1} = n\lambda R$ and $D = I$, we find that

$$Q^{-1} = I - [I + (n\lambda R)^{-1}]^{-1}. \tag{2.15}$$

Substituting (2.15) in (2.14) and factorizing we get

$$\begin{aligned} \mathrm{L}(\boldsymbol{\gamma}|\boldsymbol{y}) \;\; \propto \;\; & \exp\left\{-\frac{1}{2\sigma_e^2}[\boldsymbol{\gamma}-(I+(n\lambda R)^{-1})^{-1}(\boldsymbol{y}-T\boldsymbol{\theta})]^T \right. \\ & \left. \times\,[\boldsymbol{\gamma}-(I+(n\lambda R)^{-1})^{-1}(\boldsymbol{y}-T\boldsymbol{\theta})]\right\}. \end{aligned} \tag{2.16}$$

In this way, the BLUP for $\boldsymbol{\gamma}$ is seen to be

$$\hat{\boldsymbol{\gamma}} \;\; = \;\; \left[I+(n\lambda)^{-1}R^{-1}\right]^{-1}(\boldsymbol{y}-T\hat{\boldsymbol{\theta}}). \tag{2.17}$$

We can rewrite expression (2.17) by applying again the Sherman-Morrison-Woodsbury formula. Let $A=(n\lambda)^{-1}R^{-1}$, $B=I$, $C^{-1}=I$ and $D=I$ to obtain

$$\begin{aligned} (I+(n\lambda)^{-1}R^{-1})^{-1} &= n\lambda R-(n\lambda)^2R(I+n\lambda R)^{-1}R, \\ &= n\lambda R-(n\lambda)^2RQ^{-1}R, \end{aligned} \tag{2.18}$$

since $Q=n\lambda R+I$. Now replacing (2.18) in (2.17) we see that

$$\begin{aligned} \hat{\boldsymbol{\gamma}} &= [n\lambda R-(n\lambda)^2RQ^{-1}R](\boldsymbol{y}-T\hat{\boldsymbol{\theta}}), \\ &= n\lambda R(\boldsymbol{y}-T\hat{\boldsymbol{\theta}})-n\lambda RQ^{-1}(Q-I)(\boldsymbol{y}-T\hat{\boldsymbol{\theta}}), \end{aligned} \tag{2.19}$$

and substituting $\hat{\boldsymbol{\theta}}$ from (2.10) produces

$$\hat{\boldsymbol{\gamma}} \;\; = \;\; n\lambda RQ^{-1}[I-T(T^TQ^{-1}T)^{-1}T^TQ^{-1}]\boldsymbol{y}. \tag{2.20}$$

Finally substituting (2.10) and (2.20) into (1.3) we obtain the BLUP of $T\boldsymbol{\theta}+\boldsymbol{\gamma}$; namely,

$$\hat{\boldsymbol{y}} \;\; = \;\; [I-Q^{-1}(I+T(T^TQ^{-1}T)^{-1}T^TQ^{-1})]\boldsymbol{y}. \tag{2.21}$$

At the beginning of this derivation we considered λ and σ_e^2 fixed. Now we will obtain their estimators. Hocking (1996) showed in his book how to apply the method of REML to find estimators for the variance components σ_b^2 and σ_e^2. This method

consists of finding a suitable matrix P such that $P\boldsymbol{y}$ gives a vector whose likelihood can be separated into two independent likelihood functions: one likelihood depending on $\boldsymbol{\theta}$ and the variance components, and the other just depending on σ_e^2 and λ. Since we already have estimated the fixed effects $\boldsymbol{\theta}$, our interest is in the likelihood that only involves the variance components. This is exactly the same approach we mentioned in Chapter I when talking about the GML approach to estimate the variance components. In the next section we will show that the REML approach is equivalent to the GML method that is applied in the Bayesian model.

Let B denote an $n \times (n-m)$ matrix that satisfies

$$B^T B = I, \tag{2.22}$$

and

$$BB^T = I - T(T^T Q^{-1} T)^{-1} T^T Q^{-1}. \tag{2.23}$$

Defining

$$P = \begin{bmatrix} (T^T Q^{-1} T)^{-1} T^T Q^{-1} \\ B^T \end{bmatrix}, \tag{2.24}$$

we will then take $\boldsymbol{W} = P\boldsymbol{y}$ for

$$\boldsymbol{W} = \begin{bmatrix} (T^T Q^{-1} T)^{-1} T^T Q^{-1} \\ B^T \end{bmatrix} \boldsymbol{y} = \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix}. \tag{2.25}$$

A straightforward calculation shows that $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are independent, so the likelihood of $\boldsymbol{W}$ is given by:

$$\mathrm{L}(\boldsymbol{W}; \sigma_e^2, \lambda) \propto \frac{1}{(\sigma_e^2)^{n/2}} |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2\sigma_e^2} (\boldsymbol{W} - \boldsymbol{\mu}_W)^T \Sigma^{-1} (\boldsymbol{W} - \boldsymbol{\mu}_W) \right\} \tag{2.26}$$

with

$$\boldsymbol{\mu}_W = \begin{pmatrix} \boldsymbol{\theta} \\ \mathbf{0} \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} I + (T^T Q^{-1} T)^{-1} & \mathbf{0} \\ \mathbf{0} & \sigma_e^2 B^T Q B \end{pmatrix}. \tag{2.27}$$

Notice that the only distribution that does not involve $\boldsymbol{\theta}$ is the distribution of $\boldsymbol{w}_2$.

Denote $\Sigma_{11} = I + (T^T Q^{-1} T)^{-1}$ and $\Sigma_{22} = \sigma_e^2 B^T Q B$. Using the independence of $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ we can rewrite the likelihood of $\boldsymbol{W}$ as

$$\begin{aligned} \mathrm{L}(\boldsymbol{W}; \sigma_e^2, \lambda) \;\; \propto \;\; & \frac{1}{(\sigma_e^2)^{m/2}} \frac{1}{(\sigma_e^2)^{(n-m)/2}} \frac{1}{|\Sigma_{11}|^{m/2} |\Sigma_{22}|^{(n-m)/2}} \\ & \times \exp\left\{ \frac{1}{2\sigma_e^2} (\boldsymbol{w}_1 - \boldsymbol{\theta})^T \Sigma_{11}^{-1} (\boldsymbol{w}_1 - \boldsymbol{\theta}) + \frac{1}{2\sigma_e^2} \boldsymbol{w}_2^T \Sigma_{22}^{-1} \boldsymbol{w}_2 \right\}, \end{aligned}$$

and hence $\ell(\boldsymbol{W}) = \ell(\boldsymbol{w}_1) + \ell(\boldsymbol{w}_2)$, where ℓ denote the log-likelihood function. In particular,

$$\begin{aligned} \ell(\boldsymbol{w}_2) \;\; =_c \;\; & -\frac{(n-m)}{2} \log \sigma_e^2 - \frac{(n-m)}{2} \log |B^T Q B| - \frac{1}{2\sigma_e^2} \mathbf{w}_2^T (B^t Q B)^{-1} \mathbf{w}_2 \\ =_c \;\; & -\frac{(n-m)}{2} \log \sigma_e^2 - \frac{(n-m)}{2} \log |B^T R B + n\lambda I| \\ & -\frac{1}{2} \frac{\boldsymbol{y}^T B (B^T [R + n\lambda I] B)^{-1} B^T \boldsymbol{y}}{\sigma_e^2}, \end{aligned} \tag{2.28}$$

where "$=_c$" denotes equality up to a constant .

Fixing λ and minimizing (2.28) with respect to σ_e^2 we obtain

$$\hat{\sigma}_e^2 \;\; = \;\; \frac{\boldsymbol{y}^T B (B^T Q B)^{-1} B^T \boldsymbol{y}}{(n-m)}. \tag{2.29}$$

Substituting (2.29) in (2.28) and minimizing with respect to λ gives

$$\hat{\lambda}_{\mathrm{REML}} \;\; = \;\; \underset{\lambda}{\operatorname{argmin}} \frac{\boldsymbol{y}^T B (B^t Q B)^{-1} B^T \boldsymbol{y}}{|B^T Q B|_+^{1/(n-m)}}, \tag{2.30}$$

where $|B^T Q B|_+$ stands for the product of the non-negative eigenvalues of $B^T Q B$.

2.2 Smoothing Splines

Let $y(t_1), y(t_2), \ldots, y(t_n)$ be responses of an unknown stochastic process observed at ordinates $0 \le t_1 < \cdots < t_n \le 1$. It is common to use (1.18) to model the responses $y(\cdot)$. Often we assume that the $y(\cdot)$ mean function, $f(\cdot)$, is in

$$\begin{aligned} W_2^m[0,1] \;\; = \;\; & \{ f : [0,1] \to \Re, \; f^{(j)} \text{ are absolutely continuous} \\ & \text{for } j = 0, \ldots, (m-1), \text{ and } 0 < \int_0^1 [f^{(m)}(t)]^2 dt < \infty \}, \end{aligned}$$

where $f^{(j)}$ stands for the j^{th} derivative of f. We want to estimate $f(\cdot)$ in such a way that our estimator provides a good approximation to the data but also has a certain degree of smoothness.

Given our estimation objectives, a natural choice to fit the data can be obtained by combining the standard least-squares criterion

$$\sum_{i=1}^{n}[y(t_i) - f(t_i)]^2, \tag{2.31}$$

with the usual criterion for smoothness in $W_2^m[0,1]$, namely,

$$\int_0^1 [f^{(m)}(t)]^2 dt. \tag{2.32}$$

Using vector notation, this suggests the estimation criterion illustrated in (1.19) for the special case of $m = 2$.

Define the direct sum of two orthogonal spaces, V_1 and V_2, by $V = V_1 \bigoplus V_2$. This means that each element $x \in V$ has a unique representation $x = y + z$ with $y \in V_1$ and $z \in V_2$. Then, it can be shown that the function space $W_2^m[0,1]$ is a Hilbert space (for a more detailed discussion see Wahba, 1990; Heckman, 1997) that can be expressed as $W_2^m[0,1] = \mathcal{H}_0^m \bigoplus \mathcal{H}_1^m$, where

$$\begin{aligned} \mathcal{H}_0^m &= \{f : f^{(j)}\text{absolutely continuous,} \\ &\qquad j = 0, \ldots, (m-1),\ f^{(m)} \equiv 0\}, \end{aligned} \tag{2.33}$$

and

$$\begin{aligned} \mathcal{H}_1^m &= \{f : f^{(j)} \text{ absolutely continuous,} \\ &\qquad j = 0, \ldots, (m-1), 0 < \int_0^1 [f^{(m)}(t)]^2 dt < \infty \\ &\qquad \text{and } f^{(j)}(0) = 0, \text{ for } j = 0, \ldots, (m-1)\}. \end{aligned} \tag{2.34}$$

Now, $\mathcal{H}_0^m$ has dimension m with one set of basis functions given by $\{\phi_1(\cdot), \ldots, \phi_m(\cdot)\}$ with

$$\phi_j(t) = \frac{t^{j-1}}{(j-1)!}. \tag{2.35}$$

An inner product for $\mathcal{H}_0^m$ can be defined by

$$< f, g >_{\mathcal{H}_0^m} = \sum_{k=1}^{m} f^{(k)}(0) g^{(k)}(0)$$

for $f, g, \in \mathcal{H}_0^m$. Define the inner product of $\mathcal{H}_1^m$ by

$$< f, g >_{\mathcal{H}_1^m} = \int_0^1 f^{(m)}(t) g^{(m)}(t) dt.$$

By the orthogonality of the vector spaces, the inner product of W_2^m is just the sum of the inner products of $\mathcal{H}_0^m$ and $\mathcal{H}_1^m$. Under this inner product, W_2^m is a reproducing kernel Hilbert space with reproducing kernel $R_W = R_0 + R_1$, where

$$R_0(t, s) = \sum_{k=1}^{m} \phi_k(t) \phi_k(s) \tag{2.36}$$

and

$$R_1(t, s) = \int_0^{\min(t,s)} \frac{(t-u)^{m-1}(s-u)^{m-1}}{[(m-1)!]^2} du. \tag{2.37}$$

See, e.g., Heckman(1997).

By the Riesz representation theorem, there exists functions ξ_i such that

$$< \xi_i, f >_{\mathcal{H}_1^m} = f(t_i). \tag{2.38}$$

It can be shown (Heckman, 1997),that

$$\xi_i(t) = \int_0^{\min(t,t_i)} \frac{(t-u)^{m-1}(t_i-u)^{m-1}}{[(m-1)!]^2} du. \tag{2.39}$$

The functions $\xi_i(\cdot)$ are linearly independent and therefore they span a subspace of dimension n of $\mathcal{H}_1^m$. Moreover, it can be readily verified that the set of basis functions

$\{\phi_j\}_{j=1,m}$ and $\{\xi_i\}_{i=1,n}$ together form a basis for a space of polynomial splines of order $2m$ with knots at $t_1, \ldots, t_n$.

Given $f \in W_2^m[0,1]$ there exists a unique $\zeta \in \mathcal{H}_1^m$ such that ζ is in the orthogonal complement of span$\{\xi_i(\cdot)\}_{i=1,n}$, and we can write

$$f(t_i) = \sum_{j=1}^{m} \theta_j \phi_j(t_i) + \sum_{i=1}^{n} b_i \xi_i(t_i) + \zeta(t_i), \tag{2.40}$$

or equivalently, $\boldsymbol{f} = T\boldsymbol{\theta} + R\boldsymbol{b} + \boldsymbol{\zeta}$ with

$$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_m)^T, \text{a vector of coefficients for the basis functions in } \mathcal{H}_0^m, \tag{2.41}$$

$$\boldsymbol{b} = (b_1, \ldots, b_n)^T, \text{ a vector of coefficients for the basis functions in } \mathcal{H}_1^m, \tag{2.42}$$

$$T = \{\phi_j(t_i)\}_{\substack{i=1,n \\ j=1,m}}, \tag{2.43}$$

$$R = \{\xi_i(t_j)\}_{i,j=1,n}, \text{ and} \tag{2.44}$$

$$\boldsymbol{\zeta} = [\zeta(t_1), \ldots, \zeta(t_n)]^T.$$

So we need to find $\boldsymbol{\theta} \in \Re^m$, $\boldsymbol{b} \in \Re^n$ and ζ in the orthogonal complement of span$\{\xi_i(\cdot)\}_{i=1,n}$ such that they minimize

$$\sum_{i=1}^{n} [y(t_i) - f(t_i)]^2 + \frac{1}{n\lambda} \int_0^1 [f^{(m)}(t)]^2 dt. \tag{2.45}$$

We will show that $\zeta \equiv 0$.

The first part of the minimization criterion involves $f(t_i)$ which can be represented as

$$f(t_i) = < \xi_i, f >_{\mathcal{H}_1^m}. \tag{2.46}$$

But since $f(t_i)$ can be written as in (2.40), we can rewrite (2.46) as

$$< \xi_i, \textstyle\sum_{j=1}^{m} \theta_j \phi_j + \sum_{i=1}^{n} b_i \xi_i + \zeta >_{\mathcal{H}_1^m},$$

and this is equal to

$$< \xi_i, \textstyle\sum_{j=1}^{m} \theta_j \phi_j + \sum_{i=1}^{n} b_i \xi_i >_{\mathcal{H}_1^m},$$

since $< \xi_i, \zeta >_{\mathcal{H}_1^m} \equiv 0$. Hence, we can write the first term of the minimization criterion as

$$(\boldsymbol{y} - T\boldsymbol{\theta} - R\boldsymbol{b})^T(\boldsymbol{y} - T\boldsymbol{\theta} - R\boldsymbol{b}). \tag{2.47}$$

Now for the second term in (2.45), we have that

$$\frac{1}{n\lambda} D^{(m)} \sum_{j=1}^{m} \theta_j \phi_j(t) \;=\; 0 \tag{2.48}$$

since the ϕ_j have degree at most $(m-1)$. So

$$\begin{aligned}
\frac{1}{n\lambda}\int_0^1 \left\{ D^{(m)} \left[\sum_{j=1}^{m} \theta_j \phi_j(t) + \sum_{i=1}^{n} b_i \xi_i(t) + \zeta(t) \right] \right\}^2 dt \;&=\; \frac{1}{n\lambda}\int_0^1 \left[\sum_{i=1}^{n} b_i \xi_i^{(m)}(t) + \zeta^{(m)}(t) \right]^2 dt \\
&= \frac{1}{n\lambda} \int_0^1 \sum_{i,j=1}^{n} b_i b_j \xi_i^{(m)}(t) \xi_j^{(m)}(t) \\
&\quad + 2\frac{1}{n\lambda} \sum_{i=1}^{n} b_i \int_0^1 \xi_i^{(m)}(t) \zeta^{(m)}(t) dt \\
&\quad + \frac{1}{n\lambda} \int_0^1 [\zeta^{(m)}(t)]^2 dt \\
&= \frac{1}{n\lambda} \boldsymbol{b}^T R \boldsymbol{b} + 2\frac{1}{n\lambda} \sum_{j=1}^{n} b_j < \xi_i, \zeta > \\
&\quad + \frac{1}{n\lambda} < \zeta, \zeta > .
\end{aligned}$$

But, because of the assumptions on ζ, we know that $< \xi_i, \zeta > \equiv 0$ and $\boldsymbol{b}^T R \boldsymbol{b}$ does not depend on ζ. So, the expression is minimized when $< \zeta, \zeta > = 0$ which, in turn, implies $\zeta \equiv 0$. Therefore, the expression to minimize reduces to the Penalized Least Squares Error (PLS) criterion

$$\text{PLS} \;=\; (\boldsymbol{y} - T\boldsymbol{\theta} - R\boldsymbol{b})^T(\boldsymbol{y} - T\boldsymbol{\theta} - R\boldsymbol{b}) + \lambda \boldsymbol{b}^T R \boldsymbol{b} \tag{2.49}$$

which is now a function on $\Re^{n+m}$. More to the point, the problem has now been reduced to minimization over functions of the form $f(\cdot) = \sum_{j=1}^{m} \theta_j \phi_j(\cdot) + \sum_{i=1}^{n} b_i \xi_i(\cdot)$:

i.e., over a subspace of polynomial splines of degree $2m-1$ with knots at $t_1, \ldots, t_n$.

Now, for $r = 0, \ldots, (m-1)$, we have

$$\begin{aligned} f^{(m+r)}(t) &= D^{(m+r)} \left[\sum_{j=1}^{m} \theta_j \phi_j(t) + \sum_{i=1}^{n} b_i \xi_i(t) \right] \\ &= \sum_{j=1}^{m} \theta_j D^{(m+r)} \phi_j(t) + \sum_{i=1}^{n} b_i D^{(m+r)} \xi_i(t). \end{aligned}$$

But as we saw in (2.48), $D^{(m+r)}\phi_j(\cdot) = 0$ and therefore,

$$f^{(m+r)}(t) = \sum_{i=1}^{n} b_i D^{(m+r)} \xi_i(t).$$

Integrating the functions $\xi_i(t)$ by parts $(m-1)$ times and applying the binomial formula we see that

$$\xi_i(t) = (-1)^m \frac{(t_i - t)_+^{2m-1}}{(2m-1)!} + \frac{(-1)^m}{(2m-1)!} \sum_{k=0}^{m-1} \binom{2m-1}{k} t^k (-t_i)^{2m-k-1}.$$

Hence,

$$D^{(m+r)} \xi_i(t) = \begin{cases} 0, & \text{if} \quad t \geq t_i, \\ \frac{(t_i - t)^{m-r-1}}{(m-r-1)!}, & \text{if} \quad t < t_i, \end{cases}$$

so that

$$f^{(m+r)}(t) = \sum_{i=1}^{n} b_i \frac{(t_i - t)_+^{m-1-r}}{(m-1-r)!}.$$

For $t = 1$,

$$f^{(m+r)}(1) = 0$$

since $t_i \leq t$ for all $i = 1, \ldots, n$. For $t = 0$,

$$f^{(m+r)}(0) = \sum_{i=1}^{n} b_i \frac{(t_i)^{m-r-1}}{(m-r-1)!},$$

and therefore

$$f^{(m+r)}(0) \;=\; \sum_{i=1}^{n} b_i \phi_{m-r}(t_i) = \boldsymbol{b}^T \boldsymbol{\phi}_{m-r},$$

with $\boldsymbol{\phi}_{m-r} = [\phi_{m-r}(t_1), \ldots, \phi_{m-r}(t_n)]^T$. Thus,

$$f^{(m+r)}(0) \;=\; 0,$$

for $r = 0, \ldots, (m-1)$, if and only if

$$\boldsymbol{b}^T \boldsymbol{\phi}_{m-r} \;=\; \mathbf{0}$$

and this is true if and only if

$$T^T \boldsymbol{b} \;=\; \mathbf{0}. \tag{2.50}$$

In this way, we see that the solution of minimizing (2.49) is a natural spline of order $2m$ with knots at $t_1, \ldots, t_n$ and estimation of f is equivalent to estimation of the vectors of coefficients $\boldsymbol{\theta}$ and $\boldsymbol{b}$.

Taking derivatives with respect to $\boldsymbol{\theta}$ and $\boldsymbol{b}$ in (2.49) and setting the resulting equations equal to $\mathbf{0}$ we obtain the linear system:

$$T^T T \boldsymbol{\theta} \;=\; T^T(\boldsymbol{y} - R\boldsymbol{b}), \tag{2.51}$$

$$Q\boldsymbol{b} \;=\; n\lambda(\boldsymbol{y} - T\boldsymbol{\theta}), \tag{2.52}$$

where

$$Q \;=\; (n\lambda R + I)\,. \tag{2.53}$$

Fixing $\boldsymbol{\theta}$, we obtain

$$\hat{\boldsymbol{b}} \;=\; n\lambda Q^{-1}(\boldsymbol{y} - T\boldsymbol{\theta}). \tag{2.54}$$

Now, insert (2.54) into (2.51) to see that

$$\hat{\boldsymbol{\theta}} = (T^T Q^{-1} T)^{-1} T^T Q^{-1} \boldsymbol{y}. \tag{2.55}$$

Finally, plugging (2.55) into (2.52) we obtain

$$\begin{aligned} \hat{\boldsymbol{b}} &= n\lambda Q^{-1}\boldsymbol{y} - n\lambda Q^{-1} T (T^T Q^{-1} T)^{-1} T^T Q^{-1} \boldsymbol{y} \\ &= n\lambda Q^{-1}[I - T(T^T Q^{-1} T)^{-1} T^T Q^{-1}]\boldsymbol{y}. \end{aligned} \tag{2.56}$$

Let $\hat{\boldsymbol{f}}$ denote the estimator of $\boldsymbol{f}$. Then

$$\begin{aligned} \hat{\boldsymbol{f}} &= T\hat{\boldsymbol{\theta}} + R\hat{\boldsymbol{b}} \\ &= T(T^T Q^{-1} T)^{-1} T^T Q^{-1}\boldsymbol{y} + n\lambda R Q^{-1}[I - T(T^T Q^{-1} T)^{-1} T^T Q^{-1}]\boldsymbol{y} \\ &= [I - Q^{-1}(I - T(T^T Q^{-1} T)^{-1} T^T Q^{-1})]\boldsymbol{y}. \end{aligned} \tag{2.57}$$

Expression (2.57) is defined on the whole range $[0, 1]$, i.e., we can evaluate $\hat{\boldsymbol{f}}$ at a point t other than the design points t_i. But, when we evaluate $\hat{\boldsymbol{f}}$ only at the t_i's then (2.57) is numerically equivalent to the BLUP of $\boldsymbol{y}$ in (2.21).

Several authors have noticed the relationship between (2.57) and (2.21) and made use of it. In 1991, in a comment to a paper by Robinson (1991), Speed pointed out that smoothing splines are BLUPs. Wang (1996, 1998b) linked the smoothing spline with three particular mixed-effects models:

- **Model 1**: a model with the exact form of (2.1).
- **Model 2**: model (1.3) with $U = R$ and γ normally distributed with mean zero and variance-covariance matrix equal to R^-, the Moore-Penrose inverse of R.
- **Model 3**: model (1.3) with $U = Z$ of size $n \times n$ such that $R = ZZ^T$ and rank(R)=n. The vector of random effects $\boldsymbol{\gamma}$ (which is now of size $n \times 1$) is normally distributed with mean zero and variance-covariance matrix $\sigma_b^2 I$.

Our derivation of the smoothing spline estimator above showed that, for model 1, the smoothing spline estimator of $\boldsymbol{f}$ is the BLUP of (2.1). The last two models are just transformations of the random-effects model that yield the same answer as the smoothing spline estimator. Wang used the three models to suggest the use of existing software, like the SAS procedure proc mixed, to fit a smoothing spline.

Up to this point we have assumed that the value of the smoothing parameter λ is known. This parameter is usually unknown and there exist several criteria to estimate λ from the data. Popular methods are Generalized Cross-Validation (GCV), Unbiased Risk Prediction (UBR) and Generalized Maximum Likelihood (GML). The first two methods try to minimize the expected loss or risk function

$$\text{Risk}(\lambda) \;=\; n^{-1}E[(\boldsymbol{f}-\hat{\boldsymbol{f}})^T(\boldsymbol{f}-\hat{\boldsymbol{f}})]. \tag{2.58}$$

The UBR method selects a level of smoothing by finding a value of λ that minimizes

$$\text{UBR}(\lambda) \;=\; n^{-1}\text{RSS}(\lambda)+2n^{-1}\sigma_e^2\text{tr}(A_\lambda), \tag{2.59}$$

where tr denotes the trace of a matrix and

$$A_\lambda \;=\; I-Q^{-1}(I-T(T^TQ^{-1}T)^{-1}T^TQ^{-1}), \tag{2.60}$$

and

$$\text{RSS}(\lambda) \;=\; (\boldsymbol{y}-\hat{\boldsymbol{f}})^T(\boldsymbol{y}-\hat{\boldsymbol{f}}) \tag{2.61}$$

$\text{UBR}(\lambda)$ is an unbiased estimator of the prediction risk $\sigma_e^2+\text{Risk}(\lambda)$ (Eubank, 1988).

The GCV criterion was proposed by Craven and Wahba (1979) and it can be considered as a weighted version of the ordinary cross-validation (or leave-one-out) criterion. The GCV criterion is defined by

$$\text{GCV}(\lambda) \;=\; \frac{n^{-1}\text{RSS}(\lambda)}{[n^{-1}\text{tr}(I-A_\lambda)]^2}. \tag{2.62}$$

The GML method maximizes the likelihood of a stochastic model that will be defined in the next section and it is equivalent to obtaining the variance components via REML in the mixed-effects model. Notice that λ, in the smoothing spline setting, is the parameter that controls the trade-off between the fit of our model and our belief in the degree of smoothness of the curve whereas in the mixed-effects model setting, λ is equal to the ratio of the variance components σ_b^2/σ_e^2. The GML estimator of λ is found by minimizing the expression

$$\text{GML}(\lambda) \quad = \quad \frac{\boldsymbol{y}^T(I - A_\lambda)\boldsymbol{y}}{|I - A_\lambda|_+^{1/(n-m)}}, \tag{2.63}$$

with $|I - A_\lambda|_+$ the product of the nonzero eigenvalues of $I - A_\lambda$.

There exists other criteria to estimate λ, among them Stein's Unbiased Risk Estimator (SURE) and the Akaike Information Criterion (AIC) just to mention two. In this dissertation we will focus on GML, GCV and UBR.

2.3 Bayesian Framework

The Bayesian model considered here is the link between mixed-effects models and smoothing spline estimators. It is this framework that will allow us to use a Kalman filter algorithm (with certain assumption on the covariance structure of our model) to obtain computationally efficient estimators for any of the three different scenarios. The implementation of this algorithm will be illustrated in the chapters to follow.

Wahba (1978) showed that the fitted curve obtained in a non-parametric regression setting using smoothing splines is numerically identical to the posterior mean of an integrated Brownian motion signal plus a polynomial drift modeled using a diffuse prior. Robinson (1991) mentions a Bayesian derivation of the BLUP for (1.3) in which he considers $\boldsymbol{\theta}$ having a uniform improper prior distribution. In this section we give

a detailed derivation of Wahba's result.

Consider the model

$$y(t_i) \;=\; \sum_{j=1}^{m} \theta_j \phi_j(t_i) + \sigma_b X(t_i) + e(t_i) \tag{2.64}$$

with $X(t)$, $t \in [0,1]$, a zero-mean Gaussian stochastic process with covariance function

$$\mathrm{E}[X(t_i)X(t)] \;=\; \xi_i(t).$$

Let $\boldsymbol{X} = [X(t_1), \ldots, X(t_n)]^T$ and $\boldsymbol{e} = [e(t_1), \ldots, e(t_n)]^T$ with $\boldsymbol{e}$ distributed as $N(\mathbf{0}, \sigma_e^2 I)$ and uncorrelated with $\boldsymbol{X}$. Then, in matrix form our model becomes

$$\boldsymbol{y} \;=\; T\boldsymbol{\theta} + \sigma_b \boldsymbol{X} + \boldsymbol{e}. \tag{2.65}$$

The vector of coefficients $\boldsymbol{\theta}$ is random with a prior distribution given by $N(\mathbf{0}, \nu I)$ and such that $\boldsymbol{\theta}$ is independent of $\boldsymbol{X}$ and $\boldsymbol{e}$.

Let $\boldsymbol{f} = T\boldsymbol{\theta} + \sigma_b \boldsymbol{X}$. Given the conditions stated above, we find that the moments of the different quantities involved are:

$$\begin{aligned}
\mathrm{E}(\boldsymbol{f}) &= \mathbf{0}, \\
\mathrm{Var}(\boldsymbol{f}) &= \mathrm{E}[(T\boldsymbol{\theta} + \sigma_b \boldsymbol{X})(T\boldsymbol{\theta} + \sigma_b \boldsymbol{X})^T] \\
&= \nu T T^T + \sigma_b^2 R, \qquad (2.66) \\
\mathrm{E}(\boldsymbol{y}) &= \mathbf{0}, \\
\mathrm{E}(\boldsymbol{y}\boldsymbol{y}^T) &= \mathrm{E}[(T\boldsymbol{\theta} + \sigma_b \boldsymbol{X} + \boldsymbol{e})(T\boldsymbol{\theta} + \sigma_b \boldsymbol{X} + \boldsymbol{e})^T] \\
&= \nu T T^T + \sigma_b^2 R + \sigma_e^2 I, \qquad (2.67)
\end{aligned}$$

and

$$\begin{aligned}
\mathrm{Cov}(\boldsymbol{y}, \boldsymbol{f}) &= \mathrm{E}(\mathbf{y}\mathbf{f}^{\mathbf{T}}) \\
&= \nu T T^T + \sigma_b^2 R. \qquad (2.68)
\end{aligned}$$

This allows us to find the joint distribution of $\boldsymbol{f}$ and $\boldsymbol{y}$ as:

$$N\left(\begin{pmatrix}\mathbf{0}\\ \mathbf{0}\end{pmatrix}, \begin{pmatrix}\nu TT^T+\sigma_b^2 R & \nu TT^T+\sigma_b^2 R\\ (\nu TT^T+\sigma_b^2 R)^T & \nu TT^T+\sigma_b^2 R+\sigma_e^2 I\end{pmatrix}\right) \tag{2.69}$$

Now, applying results from multivariate analysis we see that

$$\begin{aligned}\mathrm{E}(\boldsymbol{f}|\boldsymbol{y}) &= \mathrm{Cov}(\boldsymbol{f},\boldsymbol{y})[Var(\boldsymbol{y})]^{-1}\boldsymbol{y}\\ &= (\nu TT^T+\sigma_b^2 R)(\nu TT^T+\sigma_b^2 R+\sigma_e^2 I)^{-1}\boldsymbol{y},\end{aligned} \tag{2.70}$$

and

$$\begin{aligned}\mathrm{Var}(\boldsymbol{f}|\boldsymbol{y}) &= (\nu TT^T+\sigma_b^2 R)-\\ &\quad (\nu TT^T+\sigma_b^2 R)^T(\nu TT^T+\sigma_b^2 R+\sigma_e^2 I)^{-1}(\nu TT^T+\sigma_b^2 R)\end{aligned} \tag{2.71}$$

If we then set

$$n\lambda = \frac{\sigma_b^2}{\sigma_e^2}, \tag{2.72}$$

$$\eta = \frac{\nu}{\sigma_e^2}, \tag{2.73}$$

and recall the definition of Q in (2.53), we obtain

$$\mathrm{E}(\boldsymbol{f}|\boldsymbol{y}) = (\eta TT^T+n\lambda R)(\eta TT^T+n\lambda R+I)^{-1}\boldsymbol{y} \tag{2.74}$$

Now, consider $(\eta TT^T+Q)^{-1}$. Applying again the Sherman-Morrison-Woodsbury formula, we see that

$$\begin{aligned}(\eta TT^T+Q)^{-1} &= Q^{-1}-Q^{-1}T(\eta^{-1}I+T^TQ^{-1}T)^{-1}T^TQ^{-1}\\ &= Q^{-1}-Q^{-1}T(T^TQ^{-1}T)^{-1}[\eta^{-1}(T^TQ^{-1}T)^{-1}+I]^{-1}T^TQ^{-1}.\end{aligned}$$

For η (and hence for ν) sufficiently large, the eigenvalues of $\eta^{-1}(T^TQ^{-1}T)^{-1}$ are all less than 1. So we can apply a power series expansion for matrices (Gantmakher,

1959) to obtain:

$$\begin{aligned}[I+\eta^{-1}(T^TQ^{-1}T)^{-1}]^{-1} &= \sum_{i=0}^{\infty}[\eta^{-1}(T^TQ^{-1}T)^{-1}]^i \\ &= I-\eta^{-1}(T^TQ^{-1}T)^{-1}+\eta^{-2}(T^TQ^{-1}T)^{-2}+O(\eta^{-3})\end{aligned}$$

Thus,

$$\begin{aligned}(\eta TT^T+Q)^{-1} &= Q^{-1}-Q^{-1}T(T^TQ^{-1}T)^{-1}[I-\eta^{-1}(T^TQ^{-1}T)^{-1} \\ &\quad +\eta^{-2}(T^TQ^{-1}T)^{-2}]T^TQ^{-1}+O(\eta^{-3}) \\ &= Q^{-1}-Q^{-1}T(T^TQ^{-1}T)^{-1}T^TQ^{-1}+\eta^{-1}Q^{-1}T(T^TQ^{-1}T)^{-2}T^TQ^{-1} \\ &\quad -\eta^{-2}Q^{-1}T(T^TQ^{-1}T)^{-3}T^TQ^{-1}+O(\eta^{-3}), \qquad (2.75)\end{aligned}$$

and consequently,

$$\begin{aligned}\lim_{\eta\to\infty}(\eta TT^T+n\lambda R)(\eta TT^T+Q)^{-1}\boldsymbol{y} &= \lim_{\eta\to\infty}(\eta TT^T+n\lambda R)\left\{Q^{-1}\right. \\ &\quad -Q^{-1}T(T^TQ^{-1}T)^{-1}T^TQ^{-1} \\ &\quad \left.+\eta^{-1}Q^{-1}T(T^TQ^{-1}T)^{-2}T^TQ^{-1}-O(\eta^3)\right\}\boldsymbol{y}, \\ &= T(T^TQ^{-1}T)^{-1}T^TQ^{-1}\boldsymbol{y} \\ &\quad +n\lambda RQ^{-1}[I-T(T^TQ^{-1}T)^{-1}T^TQ^{-1}]\boldsymbol{y}\end{aligned}$$

so

$$\lim_{\eta\to\infty}\mathrm{E}(\boldsymbol{f}|\boldsymbol{y}) = [I-Q^{-1}(I-T(T^TQ^{-1}T)^{-1}T^TQ^{-1})]\boldsymbol{y}. \qquad (2.76)$$

Expression (2.76) is exactly the same as expression (2.57) and (2.21). Therefore, we have three different settings: mixed-effects model, smoothing spline estimators and the Bayesian model, that provide numerically identical answers.

To estimate the smoothing parameter λ, or equivalently, the variance components σ_e^2 and σ_b^2, we use the GML method. Using the prior information on $\boldsymbol{\theta}$ we obtain the

likelihood of $\boldsymbol{y}$ and then we follow the same procedure we used when we estimated the variance components via REML.

Let B be as in (2.22) and (2.23) and define

$$P = \begin{bmatrix} \frac{1}{\sqrt{\eta}}(T^TQ^{-1}T)^{-1}T^TQ^{-1} \\ B^T \end{bmatrix}. \tag{2.77}$$

We will then take $\boldsymbol{W} = P\boldsymbol{y}$ for

$$\boldsymbol{W} = \begin{bmatrix} \frac{1}{\sqrt{\eta}}(T^TQ^{-1}T)^{-1}T^TQ^{-1} \\ B^T \end{bmatrix} \boldsymbol{y} = \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix}. \tag{2.78}$$

Then the moments of the random vector $(\boldsymbol{w}_1, \boldsymbol{w}_2)^T$ are given by

$$\mathrm{E}(\boldsymbol{W}) = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix} \quad \text{and} \quad \mathrm{Cov}(\boldsymbol{W}) = \sigma_e^2 \begin{pmatrix} I + \frac{1}{\eta}(T^TQ^{-1}T)^{-1} & \mathbf{0} \\ \mathbf{0} & B^TQB \end{pmatrix}.$$

As in (2.25), $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are independent regardless of the value of η.

Now, taking the limit when $\eta \to \infty$ we can see that the likelihood of $\boldsymbol{W}$ is given by:

$$\mathrm{L}(\boldsymbol{W}; \sigma_e^2, \lambda) \propto \left\{ \frac{1}{(\sigma_e^2)^{n/2}} |\Sigma|^{-1/2} \exp\{-\frac{1}{2\sigma_e^2} \boldsymbol{W}^T \Sigma^{-1} \boldsymbol{W} \right\},$$

where

$$\Sigma = \begin{pmatrix} I & \mathbf{0} \\ \mathbf{0} & B^TQB \end{pmatrix}.$$

Notice that in this case, due to the prior distribution of $\boldsymbol{\theta}$, the mean vector of $\boldsymbol{W}$ is just the zero vector. Again the only part of the likelihood that depends on both λ and σ_e^2 is the likelihood of $\boldsymbol{w}_1$.

In the same manner that we proceeded in (2.26), we can rewrite the likelihood

of $\boldsymbol{W}$ as

$$\begin{aligned} \mathrm{L}(\boldsymbol{W};\sigma_e^2,\lambda) \;\propto\; & \frac{1}{(\sigma_e^2)^{m/2}}\frac{1}{(\sigma_e^2)^{(n-m)/2}}\frac{1}{|I|^{m/2}|B^TQB|^{(n-m)/2}} \\ & \times\exp\left\{\frac{1}{2\sigma_e^2}\boldsymbol{w}_1^T\boldsymbol{w}_1+\frac{1}{2\sigma_e^2}\boldsymbol{w}_2^T(B^TQB)^{-1}\boldsymbol{w}_2\right\}. \end{aligned}$$

To establish the connection between our Bayesian result and (2.29) and (2.30) recall the matrix B that we used in (2.22) and (2.23). First notice that $B^TT=\mathbf{0}$ since when using (2.22) we get

$$B^TT \;=\; B^TBB^TT$$

and by applying (2.23) this is equal to

$$B^T[I-T(T^TQ^{-1}T)^{-1}T^TQ^{-1}]T.$$

We also showed that $T^T\boldsymbol{b}=0$, from (2.50) so that $\hat{\boldsymbol{b}}$ must be in the column space of B. Thus, we can write

$$\hat{\boldsymbol{b}} \;=\; B\boldsymbol{s} \tag{2.79}$$

for some vector $\boldsymbol{s}$ of length $(n-m)$. Using the normal equations (2.51) and (2.52) we obtain $T\boldsymbol{\theta}+Q\boldsymbol{b}=y$, and substituting $\hat{\boldsymbol{b}}$ in it by $B\boldsymbol{s}$ we get

$$T\boldsymbol{\theta}+QB\boldsymbol{s} \;=\; \boldsymbol{y}. \tag{2.80}$$

Multiplying both sides by B^T gives

$$B^TQB\boldsymbol{s}+B^TT\boldsymbol{\theta} \;=\; B^T\boldsymbol{y} \tag{2.81}$$

or

$$B^TQB\boldsymbol{s} \;=\; B^T\boldsymbol{y}, \tag{2.82}$$

and multiplying first by the inverse of B^TQB and secondly by B produces

$$B\boldsymbol{s} \;=\; B(B^TQB)^{-1}B^T\boldsymbol{y}. \tag{2.83}$$

Thus,

$$\hat{\boldsymbol{b}} \;=\; B(B^TQB)^{-1}B^T\boldsymbol{y}. \tag{2.84}$$

A simple calculation using $T\hat{\boldsymbol{\theta}} + R\hat{\boldsymbol{b}} = A_\lambda \boldsymbol{y}$, (2.84) and (2.54) now shows that

$$n\lambda\hat{\boldsymbol{b}} = \;\; n\lambda B(B^TQB)^{-1}B^T\boldsymbol{y} \;\; = (I - A_\lambda)\boldsymbol{y}. \tag{2.85}$$

Consequently,

$$B(B^TQB)^{-1}B^T \;\; = \;\; I - A_\lambda. \tag{2.86}$$

This shows the equivalence between the methods of REML and GML when computing σ_e^2 and λ.

One of the advantages of using the Bayesian approach is that it allows us to compute Bayesian prediction intervals. Wahba (1978, pp. 67–68) showed that

$$\mathrm{Var}(\hat{\boldsymbol{f}} - \boldsymbol{f}) \;\; = \;\; \lim_{\eta\to\infty} \sigma_e^2[(I - M)(\nu TT^T + \sigma_b^2 R)(I - M^T) + MM^T], \tag{2.87}$$

where $M = (\eta TT^T + n\lambda R)(\eta TT^T + n\lambda R + I)^{-1}$.

Expanding (2.87) we obtain

$$\begin{aligned}
\mathrm{Var}(\hat{\boldsymbol{f}} - \boldsymbol{f}) \;\; &= \;\; \lim_{\eta\to\infty} \sigma_e^2[(\eta TT^T + R) - M(\eta TT^T + R) \\
&\qquad (\eta TT^T + R)M^T + M(\eta TT^T + R)M^T] \\
&= \;\; \sigma_e^2 \lim_{\eta\to\infty}[I - Q^{-1} + Q^{-1}T(T^TQ^{-1}T)^{-1}T^TQ^{-1} + O(1/\eta)] \\
&= \;\; \sigma_e^2 A_\lambda.
\end{aligned} \tag{2.88}$$

In this way, a $(1-\alpha)100\%$ confidence interval for $f(t_i)$ is given by

$$\hat{f}(t_i) \quad \pm \quad z_{1-\alpha/2}\sqrt{\hat{\sigma}_e^2 a_{ii}}, \tag{2.89}$$

with a_{ii} the i^{th} diagonal element of A_λ and $\hat{\sigma}_e^2$ one of the usual estimators of σ_e^2 like the one obtained by ML or the one defined by (1.14).

2.4 Synopsis

We have shown that the estimates of the mixed-effects model (1.3), with $U \equiv I$, evaluated at the design points t_i are numerically the same as the smoothing splines fitted values in (1.18), and the posterior mean of a signal which is a stochastic process plus a polynomial trend with improper prior (1.22). Estimation of the variance components is done via REML in the mixed-effects model setting and via GML in the Bayesian model. We have illustrated the equivalence between these two methods. Based on these results, we can apply any of the procedures to estimate any of the quantities required. The choice of the estimation procedure to use then will depend entirely on computational efficiency or software availability.

CHAPTER III

THE THREE TOOLS THEOREM

In the previous chapter we showed the numerical equivalence between the estimators obtained using three different methods: a particular mixed-effects model (1.3) with $U \equiv I$, the smoothing splines estimator (2.49), and the Bayesian model (2.64). These relationships are not new and they have been around for some time now.

Wahba was aware of the relationship between the smoothing spline estimator and the Bayesian model and, together with Kimeldorf (1970), she proved that corresponding predictions at particular values t_i, $i = 1, \ldots, n$, are BLUP. Wahba (1978) used the connection between the Bayesian model and smoothing spline estimators to obtain "Bayesian" confidence intervals for the function f evaluated at the design points. Speed (1991) remarked that "smoothing splines are BLUP" and pondered the coverage properties of the "Bayesian" posterior intervals when using Wahba's "Bayesian" approach. One other point made by Speed that didn't catch as much attention as his remark about smoothing splines and BLUPs was his observation about penalized least squares being also BLUP.

Speed observed that if we have two vectors of fixed effects, $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$, we could try to estimate them by applying the method of R-weighted LS, plus a penalty on the vector $\boldsymbol{\gamma}$, and we will obtain the same solutions as the BLUP of (1.3). However, the relationship between the mixed-effects model, the Bayesian model and smoothing splines estimators (and even more generally, penalized least squares) has not been thoroughly exploited.

3.1 Main Theorem

The following theorem connects the mixed-effects model, the Bayesian model and penalized least squares in a general context that will allow us to take advantage of some of the theoretical and computational results and interpretations of the other two.

Theorem 3.1.1 *Consider the linear mixed-effects model*

$$\boldsymbol{y} = T\boldsymbol{\theta} + U\boldsymbol{\gamma} + \boldsymbol{e}, \tag{3.1}$$

with $\boldsymbol{y}$ *an* $n \times 1$ *vector of responses,* T *and* U *design matrices for the fixed and random effects of dimensions* $n \times m$ *and* $n \times q$*, respectively. Let* $\boldsymbol{\theta}$ *denote an* $m \times 1$ *vector of fixed effects,* $\boldsymbol{\gamma}$ *a* $q \times 1$ *vector of random effects, and* $\boldsymbol{e}$ *an* $n \times 1$ *vector of random errors which are normally distributed with zero mean and variance-covariance matrix* $\sigma_e^2 I$ *and uncorrelated with* $\boldsymbol{\gamma}$*. Also,* $\boldsymbol{\gamma}$ *is normally distributed with moments given by*

$$E(\boldsymbol{\gamma}) = \mathbf{0},$$

and

$$Var(\boldsymbol{\gamma}) = \sigma_b^2 R.$$

Let

$$Q = (UR_\lambda U^T + I), \tag{3.2}$$

with

$$R_\lambda = n\lambda R, \tag{3.3}$$

and

$$n\lambda = \frac{\sigma_b^2}{\sigma_e^2}. \tag{3.4}$$

Then, the BLUP of $T\boldsymbol{\theta} + U\boldsymbol{\gamma}$*, given by*

$$\hat{\boldsymbol{y}} \;=\; I - Q^{-1}[I - T(T^TQ^{-1}T)^{-1}T^TQ^{-1}]\boldsymbol{y}, \tag{3.5}$$

is numerically the same solution as **1.** *and* **2.** *with*

1.

$$\lim_{\nu\to\infty} \mathrm{E}(T\boldsymbol{\theta} + U\boldsymbol{\gamma}|\boldsymbol{y}) \tag{3.6}$$

for the Bayesian model

$$\boldsymbol{y} \;=\; T\boldsymbol{\theta} + U\boldsymbol{\gamma} + \boldsymbol{e}, \tag{3.7}$$

where T*,* U*,* $\boldsymbol{\gamma}$*, and* $\boldsymbol{e}$ *are as before, the vector* $\boldsymbol{\theta}$ *is normally distributed with mean zero, variance-covariance matrix* νW*, for some positive-definite matrix* W*, and the vectors* $\boldsymbol{\theta}$*,* $\boldsymbol{\gamma}$*,* $\boldsymbol{e}$ *are independent of each other,*

and

2. *the solution* $\hat{\boldsymbol{f}} = T\hat{\boldsymbol{\theta}} + U\hat{\boldsymbol{\gamma}}$ *obtained by minimizing the Penalized Least Squares error criterion*

$$\mathrm{PLS}(\boldsymbol{\theta},\boldsymbol{\gamma}) \;=\; (\boldsymbol{y} - T\boldsymbol{\theta} - U\boldsymbol{\gamma})^T(\boldsymbol{y} - T\boldsymbol{\theta} - U\boldsymbol{\gamma}) + \boldsymbol{\gamma}^T R_\lambda^{-1}\boldsymbol{\gamma}, \tag{3.8}$$

with respect to $\boldsymbol{\theta}$ *and* $\boldsymbol{\gamma}$*.*

Proof. Under model (3.1), the moments of $\boldsymbol{y}$ are given by

$$\mathrm{E}(\boldsymbol{y}) \;=\; T\boldsymbol{\theta}, \tag{3.9}$$

and

$$\mathrm{Var}(\boldsymbol{y}) \;=\; \sigma_b^2 URU^T + \sigma_e^2 I. \tag{3.10}$$

Let

$$Q \;=\; (R_\lambda + I)\,. \tag{3.11}$$

Then, using the distribution of $\boldsymbol{y}$ given $\boldsymbol{\gamma}$ and the distribution of $\boldsymbol{\gamma}$, we find that the joint density of $\boldsymbol{y}$ and $\boldsymbol{\gamma}$ is

$$\begin{aligned} \mathrm{L}(\boldsymbol{y},\boldsymbol{\gamma}) &\propto \exp\left\{-\frac{1}{2\sigma_e^2}(\boldsymbol{y}-T\boldsymbol{\theta}-U\boldsymbol{\gamma})^T(\boldsymbol{y}-T\boldsymbol{\theta}-U\boldsymbol{\gamma})\right. \\ &\quad \left.-\frac{1}{2\sigma_e^2}\boldsymbol{\gamma}^T R^{-1}\boldsymbol{\gamma}\right\}, \end{aligned} \tag{3.12}$$

and we can obtain the normal equations derived by Henderson (1959) by differentiating with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$: namely,

$$T^TT\boldsymbol{\theta}+T^TU\boldsymbol{\gamma} = T^T\boldsymbol{y} \tag{3.13}$$

$$U^TT\boldsymbol{\theta}+(U^TU+R_\lambda^{-1})\boldsymbol{\gamma} = U^T\boldsymbol{y}. \tag{3.14}$$

To eliminate $\boldsymbol{\gamma}$ from (3.14), pre-multiply by $U(U^TU+R_\lambda^{-1})^{-1}$ to obtain

$$T[I-U(U^TU+R_\lambda^{-1})^{-1}U^T]T\boldsymbol{\theta} = T^T[I-U(U^TU+R_\lambda^{-1})^{-1}U^T]\boldsymbol{y}. \tag{3.15}$$

Using (3.11) and the Sherman-Morrison-Woodsbury formula we have

$$Q^{-1} = I-U(U^TU+R_\lambda^{-1})^{-1}U^T \tag{3.16}$$

and substituting (3.16) in (3.15) gives

$$\hat{\boldsymbol{\theta}} = (T^TQ^{-1}T)^{-1}T^TQ^{-1}\boldsymbol{y}. \tag{3.17}$$

Plugging (3.17) into (3.14) produces

$$U^TT(T^TQ^{-1}T)^{-1}T^TQ^{-1}\boldsymbol{y}+(U^TU+R_\lambda^{-1})\boldsymbol{\gamma} = U^T\boldsymbol{y},$$

so

$$U\hat{\boldsymbol{\gamma}} = [I-T(T^TQ^{-1}T)^{-1}T^TQ^{-1}-Q^{-1}+Q^{-1}T(T^TQ^{-1}T)^{-1}T^TQ^{-1}]\boldsymbol{y}. \tag{3.18}$$

In this way, the predicted values of $T\boldsymbol{\theta}+U\boldsymbol{\gamma}$ are given by

$$\hat{\boldsymbol{y}} = \{I-Q^{-1}[I-T(T^TQ^{-1}T)^{-1}T^TQ^{-1}]\}\boldsymbol{y}. \tag{3.19}$$

To show that minimization of the PLS criterion produces the same numerical result as the BLUP for the mixed-effects model (3.1), we differentiate PLS($\boldsymbol{\theta}, \boldsymbol{\gamma}$) with respect to $\boldsymbol{\gamma}$ and obtain

$$\hat{\boldsymbol{\gamma}} = [U^T U + R_\lambda^{-1}]^{-1} U^T (\boldsymbol{y} - T\boldsymbol{\theta}). \tag{3.20}$$

Differentiating (3.8) with respect to $\boldsymbol{\theta}$ and equating to zero gives

$$\begin{aligned} \frac{\partial \text{PLS}(\boldsymbol{\theta}, \boldsymbol{\gamma})}{\partial \boldsymbol{\theta}} &= -T^T(\boldsymbol{y} - T\boldsymbol{\theta} - U\boldsymbol{\gamma}) \\ &= -T^T\boldsymbol{y} + T^T T\boldsymbol{\theta} + T^T U \left[U^T U + R_\lambda^{-1}\right]^{-1} U^T \boldsymbol{y} \\ &\quad -T^T U \left[U^T U + R_\lambda^{-1}\right]^{-1} U^T T\boldsymbol{\theta} \\ &= 0 \end{aligned}$$

or

$$T^T \left\{ I - U \left[U^T U + R_\lambda^{-1}\right]^{-1} U^T \right\} T\boldsymbol{\theta} = T^T \left\{ I - U \left[U^T U + R_\lambda^{-1}\right]^{-1} U^T \right\} \boldsymbol{y}.$$

Applying (3.16) again then gives

$$T^T Q^{-1} T\boldsymbol{\theta} = T^T Q^{-1} \boldsymbol{y}$$

or

$$\hat{\boldsymbol{\theta}} = (T^T Q^{-1} T)^{-1} T^T Q^{-1} \boldsymbol{y}. \tag{3.21}$$

Finally, substituting (3.20) and (3.21) in (**??**) we have

$$\begin{aligned} \hat{\boldsymbol{f}} &= T\hat{\boldsymbol{\theta}} + U\hat{\boldsymbol{\gamma}} \\ &= T(T^T Q^{-1} T)^{-1} T^T Q^{-1} \boldsymbol{y} + U[U^T U + R_\lambda^{-1}]^{-1} U^T [\boldsymbol{y} - T(T^T Q^{-1} T)^{-1} T^T Q^{-1}]. \end{aligned}$$

Notice that $U[U^T U + R_\lambda^{-1}]^{-1} U^T = I - Q^{-1}$ and hence

$$\hat{\boldsymbol{f}} = \{I - Q^{-1}[I - T(T^T Q^{-1} T)^{-1} T^T Q^{-1}]\} \boldsymbol{y}, \tag{3.22}$$

which is the same as (3.19).

It remains to show that under the Bayesian model with diffuse prior, $E(T\boldsymbol{\theta} + U\boldsymbol{\gamma}|\boldsymbol{y})$ agrees with (3.19) and (3.22). In this case, the joint distribution of $T\boldsymbol{\theta} + U\boldsymbol{\gamma}$ and $\boldsymbol{y}$ is found to be normal with mean vector

$$\boldsymbol{\mu} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \tag{3.23}$$

and a variance-covariance matrix

$$\begin{aligned} \Sigma &= \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \\ &= \begin{pmatrix} \nu TTT + n^{-1}\sigma_b^2 URU^T & \nu TT^T + n^{-1}\sigma_b^2 URU^T \\ (\nu TT^T + n^{-1}\sigma_b^2 UU^T)^T & \nu TT^T + n^{-1}\sigma_b^2 UU^T + \sigma_e^2 I \end{pmatrix}. \end{aligned} \tag{3.24}$$

So,

$$\begin{aligned} \mathrm{E}(T\boldsymbol{\theta} + U\boldsymbol{\gamma}|\boldsymbol{y}) &= \mathrm{Cov}(T\boldsymbol{\theta} + U\boldsymbol{\gamma}, \boldsymbol{y})[Var(\boldsymbol{y})]^{-1}\boldsymbol{y} \\ &= (\nu TT^T + n^{-1}\sigma_b^2 URU^T) \\ &\quad \times(\nu TT^T + n^{-1}\sigma_b^2 URU^T + \sigma_e^2 I)^{-1}\boldsymbol{y}, \end{aligned} \tag{3.25}$$

and

$$\mathrm{Var}(T\boldsymbol{\theta} + U\boldsymbol{\gamma}|\boldsymbol{y}) = (\nu TT^T + n^{-1}\sigma_b^2 URU^T) - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}, \tag{3.26}$$

where

$$\begin{aligned} \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} &= (\nu TT^T + n^{-1}\sigma_b^2 URU^T)^T(\nu TT^T + n^{-1}\sigma_b^2 URU^T + \sigma_e^2 I)^{-1} \\ &\quad \times(\nu TT^T + n^{-1}\sigma_b^2 URU^T). \end{aligned}$$

Parameterizing in the same manner we did in Chapter II for (2.72) and (2.73) and taking the limit as η tends to infinity we obtain exactly the same expression as in

(3.19) and (3.22). $\diamond$

The proof of theorem (3.1.1) implicity assumes that the variance-covariance matrix of the random effects, R, is invertible and that the design matrices U and T are full column rank, in the mixed-effects or Bayesian model setting. If we deal with the PLS case for spline smoothing and if we have distinct design points t_i, then the matrix with elements of the form

$$R = \left\{ \int_0^{\min(t_i,t_j)} \frac{(t_i - u)^{m-1}(t_j - u)^{m-1}}{[(m-1)!]^2} du \right\}_{i,j=1,n}$$

is invertible but this is not true if we have repeated observations for some t_i's.

Suppose that, in the PLS setting, the matrix R is not invertible. In this situation, the matrix $Q = I + n\lambda R$ will still be invertible. Thus, our only concern is that the matrix T is less than full rank and. To deal with this case, we can use conditional inverses (as defined by Graybill, 1976, pp. 31).

Recall that the form of our penalized least squares estimator is

$$\hat{\boldsymbol{f}} \;=\; I - Q^{-1}[I - T(T^T Q^{-1} T)^{-1} T^T Q^{-1}]\boldsymbol{y}.$$

If T is not full column rank, then $T^T Q^{-1} T$ is not invertible, but any $T(T^T Q^{-1} T)^c T^T$ will be invariant with respect to the choice of conditional inverse $(T^T Q^{-1} T)^c$. We will proof this in the next corollary that follows from Theorem 1.5.25 of Grayville (1976, pp. 33).

Corollary 3.1.2 *For any matrix T of size n by m and rank $r > 0$ the expression $T(T^T Q^{-1} T)^c T^T$ is invariant for any c-inverse of $T^T Q^{-1} T$.*

Proof. Let A and B be two c-inverses of $T^T Q^{-1} T$, and let $T = T_L T_R$ be the full rank decomposition of T. Then

$$(T^T Q^{-1} T) A (T^T Q^{-1} T) \;=\; (T^T Q^{-1} T) B (T^T Q^{-1} T),$$

by the definition of c-inverse. Replace T in the expression above by its full rank decomposition to get

$$T_R^T T_L^T Q^{-1} T_L T_R A T_R^T T_L^T Q^{-1} T_L T_R = T_R^T T_L^T Q^{-1} T_L T_R B T_R^T T_L^T Q^{-1} T_L T_R.$$

Now, multiply both sides on the left by $(T_R^T)^c$ and on the right by $(T_R)^c$ to obtain

$$T_L^T Q^{-1} T_L T_R A T_R^T T_L^T Q^{-1} T_L = T_L^T Q^{-1} T_L T_R B T_R^T T_L^T Q^{-1} T_L.$$

Multiplying both sides of the equation, on the left and on the right by $(T_L^T Q^{-1} T_L)^{-1}$ gives

$$T_R A T_R^T = T_R B T_R^T,$$

and finally, multiply on the left by T_L and on the right by T_L^T produces

$$T A T^T = T B T^T.$$

$\diamond$

So this corollary tell us that, in the PLS setting, we don't need to worry about having a non-invertible matrix R or a matrix T with less than full column rank.

3.2 Estimation of $\boldsymbol{\lambda}$, and the Variance Components

The second part of this chapter is dedicated to justifying the use of the GCV, GML and UBR methods for estimation of the variance components and/or the smoothing parameter. Under the mixed-effects model setting we know that we can use the method of GML or REML to find the smoothing parameter $\hat{\lambda}_{\text{GML}}$ and $\hat{\sigma}^2_{e\text{GML}}$, given by (2.63) and (2.29) respectively, and substituting these values in (2.6) we can obtain $\hat{\sigma}^2_{b\text{GML}}$. On the other hand, we can use the penalized least squares approach and try

to find the variance components via GCV or UBR (if the true value of σ_e^2 is known for the later criterion).

To be able to use these criteria, we need to show that the GML, GCV and UBR functions are minimized by the ratio of the variance components. The following lemma is the generalization of Theorem 5.6 (Eubank, 1988, pp.244-247).

Lemma 3.2.1 *Let* E *and* $\mathrm{E}_{\boldsymbol{\gamma}}$ *denote the expectation with respect to the distribution of* $\boldsymbol{e}$ *and* $\boldsymbol{\gamma}$*, respectively. Then* $\mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[\mathrm{GCV}(\lambda)]$, $\mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[\mathrm{GML}(\lambda)]$ *and* $\mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[\mathrm{UBR}(\lambda)]$ *are all minimized at* $\lambda = \sigma_b^2/\sigma_e^2$.

Proof. Since the GCV and UBR criteria both depend on RSS we will first derive $\mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[RSS(\lambda)]$. We know that

$$\mathrm{RSS}(\lambda) \;=\; \boldsymbol{y}^T(I - A_\lambda)\boldsymbol{y}. \tag{3.27}$$

Taking expectation of $\mathrm{RSS}(\lambda)$ with respect to $\boldsymbol{e}$ we get

$$\mathrm{E}[RSS(\lambda)] \;=\; \mathrm{tr}[\sigma_e^2(I - A_\lambda)^2] + \boldsymbol{f}^T(I - A_\lambda)^2\boldsymbol{f}. \tag{3.28}$$

Now, taking expectation with respect to $\boldsymbol{\gamma}$ we obtain

$$\mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[RSS(\lambda)] \;=\; \mathrm{tr}[\sigma_e^2(I - A_\lambda)^2] + \mathrm{tr}[\sigma_b^2(I - A_\lambda)^2 URU^T], \tag{3.29}$$

since $(I - A_\lambda)T = \mathbf{0}$ and $\mathrm{E}(\gamma) = [0]$.

Let $\lambda_o = \sigma_b^2/n\sigma_e^2$, then

$$\mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[RSS(\lambda)] \;=\; \sigma_e^2\mathrm{tr}[(I - A_\lambda)^2] + \sigma_e^2 n\lambda_o\mathrm{tr}[(I - A_\lambda)^2 URU^T]. \tag{3.30}$$

Writing $URU^T = (1/n\lambda)(Q - I)$ we can see that

$$\begin{aligned}
\mathrm{tr}[(I - A_\lambda)^2 URU^T] &= \frac{1}{n\lambda}\mathrm{tr}[(I - A_\lambda)Q(I - A_\lambda)] \\
&= \frac{1}{n\lambda}\mathrm{tr}[\{I - Q^{-1}T(T^TQ^{-1}T)^{-1}T^T - (I - A_\lambda)\}(I - A_\lambda)] \\
&= \frac{1}{n\lambda}\mathrm{tr}[(I - A_\lambda) - (I - A_\lambda)^2],
\end{aligned} \tag{3.31}$$

and substituting (3.31) into (3.28)

$$\mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[RSS(\lambda)] = \sigma_e^2\mathrm{tr}[(I-A_\lambda)^2] + \sigma_e^2 n\lambda_o \mathrm{tr}[(I-A_\lambda)-(I-A_\lambda)^2]. \quad (3.32)$$

Now, we have shown that it is possible to choose a matrix B that satisfies $B^TB = I$, $BB^T = I - T(T^TQ^{-1}T)^{-1}T^TQ^{-1}$, and $B^TT = 0$, so we can write

$$\begin{aligned} B^TQB &= B^T(n\lambda URU^T + I)B \\ &= n\lambda B^TURU^TB + I. \end{aligned} \quad (3.33)$$

Let

$$\Lambda = \mathrm{diag}\{d_1, \ldots, d_{n-m}\} \quad (3.34)$$

be the matrix of eigenvalues for B^TURU^TB with corresponding matrix of eigenvectors V. Then, we can write

$$B^TQB = V(n\lambda\Lambda + I)V^T. \quad (3.35)$$

Using our representation for B^TQB we can show that

$$\mathrm{tr}[(I-A_\lambda)] = \sum_{i=1}^{n-m}(n\lambda d_i + 1)^{-1} \quad (3.36)$$

and hence

$$\mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[RSS(\lambda)] = \sigma_e^2\sum_{i=1}^{n-m}\frac{(n\lambda_o d_i + 1)}{(n\lambda d_i + 1)}. \quad (3.37)$$

Replacing (3.37) into $\mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[\mathrm{GCV}(\lambda)]$ we have

$$\begin{aligned} \mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[\mathrm{GCV}(\lambda)] &= \frac{n\mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[RSS(\lambda)]}{[\mathrm{tr}(I-A_\lambda)]^2} \\ &= \frac{n\sigma_e^2\sum_{i=1}^{n-m}(n\lambda_o d_i+1)(n\lambda d_i+1)^{-2}}{[\sum_{i=1}^{n-m}(n\lambda d_1+i)^{-1}]^2}. \end{aligned} \quad (3.38)$$

When $\lambda = \lambda_o$ expression (3.38) reduces to

$$\mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[\mathrm{GCV}(\lambda_o)] \;=\; \frac{\sigma_e^2}{\sum_{i=1}^{n-m}(n\lambda d_1 + i)^{-1}}. \tag{3.39}$$

Thus, the value of λ_o will minimize the GCV criterion if and only if

$$\left[\sum_{i=1}^{n-m}(n\lambda_o d_i + 1)(n\lambda d_i + 1)^{-2}\right]\left[\sum_{j=1}^{n-m}(n\lambda_o d_j + 1)^{-1}\right] - \left[\sum_{i=1}^{n-m}(n\lambda d_i + 1)^{-1}\right]^2 \geq 0.$$

A direct application of the Cauchy-Schwartz inequality shows that

$$\left[\sum_{i=1}^{n-m}(n\lambda d_i + 1)^{-1}\right]^2 \;\leq\; \left[\sum_{i=1}^{n-m}\frac{(n\lambda_o d_i + 1)}{(n\lambda d_i + 1)^2}\right]\left[\sum_{j=1}^{n-m}(n\lambda_o d_j + 1)^{-1}\right], \tag{3.40}$$

from which the result follows.

We prove next that the minimization of the UBR criterion is attained at $\lambda = \lambda_o$. The UBR criterion is given by

$$\mathrm{UBR}(\lambda) \;=\; \frac{1}{n}\boldsymbol{y}^T(I - A_\lambda)^2\boldsymbol{y} + \frac{2}{n}\sigma_e^2\mathrm{tr}(A_\lambda). \tag{3.41}$$

The expectation of UBR with respect first to the errors and secondly to the random effects $\boldsymbol{\gamma}$ is

$$\begin{aligned}\mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[\mathrm{UBR}(\lambda)] &= \frac{1}{n}\mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[\mathrm{RSS}(\lambda)] + \frac{2\sigma_e^2}{\mathrm{tr}}(A_\lambda) \\ &= \frac{\sigma_e^2}{n}\sum_{i=1}^{n-m}\frac{(n\lambda_o d_i + 1)}{(n\lambda d_i + 1)} + \frac{2\sigma_e^2}{\mathrm{tr}}(A_\lambda).\end{aligned} \tag{3.42}$$

We can write A_λ as $-[(I - A_\lambda) - I]$, and this allow us to find that its eigenvalues are equal to $1 - 1/(n\lambda d_i + 1)$. Hence

$$\mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[\mathrm{UBR}(\lambda)] \;=\; \frac{\sigma_e^2}{n}\sum_{i=1}^{n-m}\frac{(n\lambda_o d_i + 1)}{(n\lambda d_i + 1)} + \frac{2\sigma_e^2}{\sum}(_{i=1}^{n-m}\left(1 - \frac{1}{(n\lambda d_i + 1)}\right) \tag{3.43}$$

with λ_o as before. Differentiating the expectation with respect to λ we get

$$\frac{\partial \mathrm{E}_{\boldsymbol{\gamma}}\mathrm{E}[\mathrm{UBR}(\lambda)]}{\partial\lambda} \;=\; \frac{\sigma_e^2}{n}\left[-2\sum_{i=1}^{n-m}\frac{(nd_i)(n\lambda_o d_i + 1)}{(n\lambda d_i + 1)^3} + 2\sum_{i=1}^{n-m}\frac{nd_i}{(n\lambda d_i + 1)^2}\right] \tag{3.44}$$

and this expression is zero when $\lambda = \lambda_o$. Differentiating again, we get

$$\frac{\partial^2 \mathrm{E}_{\boldsymbol{\gamma}} \mathrm{E}[\mathrm{UBR}(\lambda)]}{\partial \lambda^2} = \frac{\sigma_e^2}{n}\left[6\sum_{i=1}^{n-m}\frac{(nd_i)^2(n\lambda_o d_i+1)}{(n\lambda d_i+1)^4} - 4\sum_{i=1}^{n-m}\frac{(nd_i)^2}{(n\lambda d_i+1)^3}\right],$$

which is

$$\frac{\sigma_e^2}{n} 2\sum_{i=1}^{n-m}\frac{(nd_i)^2}{(n\lambda d_i+1)^3} \geq 0$$

at $\lambda = \lambda_o$ showing that λ_o minimizes the UBR criterion.

Finally, we will prove that λ_o is a minimizer of the GML(λ) criterion. The GML criterion is given by

$$\mathrm{GML}(\lambda) = \frac{\boldsymbol{y}^T(I-A_\lambda)\boldsymbol{y}}{|I-A_\lambda|_+^{1/(n-m)}}.$$

Proceeding as in the proofs for the GCV and UBR criteria, we can show that

$$\mathrm{E}_{\boldsymbol{\gamma}} \mathrm{E}[\mathrm{GML}(\lambda)] = \frac{\sigma_e^2 \mathrm{tr}[(I-A_\lambda)] + n\lambda_o \mathrm{tr}[(I-A_\lambda)(Q-I)]}{[\prod_{i=1}^{n-m}(n\lambda d_i+1)^{-1/(n-m)}]}. \tag{3.45}$$

By the cyclic property of the trace, we can write $\mathrm{tr}[(I-A_\lambda)(Q-I)]$ as

$$\begin{aligned}
\mathrm{tr}[(Q-I)(I-A_\lambda)] &= [I - T(T^TQ^{-1}T)^{-1}T^TQ^{-1} - (I-A_\lambda)] \\
&= [BB^T - B(B^TQB)^{-1}B^T] \\
&= B[I-(B^TQB)^{-1}]B^T,
\end{aligned}$$

since $BB^T = I - T(T^TQ^{-1}T)^{-1}T^TQ^{-1}$. Hence

$$\begin{aligned}
\sigma_e^2 \mathrm{tr}[(I-A_\lambda)] + n\lambda_o \mathrm{tr}[(I-A_\lambda)(Q-I)] &= \sum_{i=1}^{n-1}\frac{1}{(n\lambda d_i+1)} + \frac{\lambda_o}{\lambda}\sum_{i=1}^{n-m}\frac{n\lambda d_i}{(n\lambda d_i+1)} \\
&= \sum_{i=1}^{n-m}\frac{(n\lambda_o d_i+1)}{(n\lambda d_i+1)}.
\end{aligned} \tag{3.46}$$

Replacing (3.46) in (3.45) produces

$$\mathrm{E}_{\boldsymbol{\gamma}} \mathrm{E}[\mathrm{GML}(\lambda)] = \frac{\sigma_e^2}{\prod_{i=1}^{n-m}(n\lambda d_i+1)^{-1/(n-m)}}\sum_{i=1}^{n-m}\frac{(n\lambda_o d_i+1)}{(n\lambda d_i+1)}. \tag{3.47}$$

Now, taking the logarithm of the expectation we obtain

$$\begin{aligned} \log \mathrm{E}_{\boldsymbol{\gamma}} \mathrm{E}[\mathrm{GML}(\lambda)] &= \log\sigma_e^2 + \frac{1}{n-m}\sum_{i=1}^{n-m}\log(n\lambda d_i + 1) \\ &\quad + \log\left\{\sum_{i=1}^{n-m}\frac{(n\lambda_o d_i + 1)}{(n\lambda d_i + 1)}\right\}, \end{aligned} \tag{3.48}$$

and substituting λ_o for λ, we get

$$\begin{aligned} \log \mathrm{E}_{\boldsymbol{\gamma}} \mathrm{E}[\mathrm{GML}(\lambda_o)] &= \log\sigma_e^2 + \frac{1}{n-m}\sum_{i=1}^{n-m}\log(n\lambda_o d_i + 1) \\ &\quad + \log(n-m). \end{aligned} \tag{3.49}$$

Following analogous steps to the ones taken to prove the minimization of GCV by λ_o, we take the difference of logarithms of the expectations to see that a sufficient condition for minimization at λ_o is that

$$\log\left[\frac{1}{n-m}\sum_{i=1}^{n-m}\frac{(n\lambda_o d_i + 1)}{(n\lambda d_i + 1)}\right] - \frac{1}{(n-m)}\sum_{i=1}^{n-m}\log\left[\frac{(n\lambda_o d_i + 1)}{n\lambda d_i + 1}\right] \geq 0.$$

The logarithm is a concave function so we can now apply Jensen's inequality to show that

$$\log\left[\frac{1}{n-m}\sum_{i=1}^{n-m}\frac{(n\lambda_o d_i + 1)}{(n\lambda d_i + 1)}\right] \geq \frac{1}{(n-m)}\sum_{i=1}^{n-m}\log\left[\frac{(n\lambda_o d_i + 1)}{n\lambda d_i + 1}\right]$$

and this expression is zero at $\lambda = \lambda_o$.

$\diamond$

3.3 Simulations

There exists several studies comparing the GCV and GML estimates of λ in the smoothing splines framework. Wahba (1985) studied the asymptotic behavior of both smoothing parameter estimates and validated it with a Monte Carlo simulation. She found theoretically and with the simulation that the GCV estimate of λ performs

better than its GML counterpart. She used periodic smoothing splines with $m = 2$, 4 different error variances, ranging from .0125 to .020, and 3 different functions. The simulations were done with sample sizes of 32, 64 and 128. Khon, Ansley and Tharm (1991) conducted a larger simulation study with 10 functions that included light, normal and heavy tail distributions. They also considered equally and unequally spaced data points and had a wider range of variability. They found that both GML and GCV estimators perform in similar ways, although when using splines with $m = 3$ and considering the function and its first derivative, the GML estimate outperforms the GCV estimate.

We conducted a small simulation study to assess and compare the performance of the GCV and GML estimates of the variance components in the mixed-effects model setting. Consider the model

$$y_{ij} \quad = \quad \theta + \gamma_i + e_{ij} \tag{3.50}$$

with $i = 1, \ldots, M$ and $j = i, \ldots, n_i$. The response y_{ij} corresponds to the jth observation from the ith subject. The fixed parameter θ is the mean across subjects, the γ_i's are the random effects corresponding to subject i and they are independently normally distributed with zero mean and variance σ_b^2. The errors, e_{ij}, are independent normally distributed random variables with zero mean and variance σ_e^2 and independent of the γ_i's.

For simplicity, we will consider a balance design: i.e., $n_1 = n_2 = \cdots = n_6$, with $M = 6$ subjects and 3 and 10 replications per subject, producing a total number of observations equal to 18 and 60 respectively. We simulated the two settings a 1000 times and we then studied three cases:

1. The random-effect to noise ratio is larger than 1. We chose values of $\sigma_b = 24$ and $\sigma_e = 4$.

2. The random-effect to noise ratio is smaller than 1. We chose values of $\sigma_b = 4$ and $\sigma_e = 24$.

3. The random-effect to noise ratio is equal to 1. We chose values of $\sigma_b = 5$ and $\sigma_e = 5$.

We used the function lme in R which gives REML estimates for σ_e^2 and σ_b^2 (remember that REML and GML were proven to be equivalent in Chapter II). We are calling these estimates respectively $\hat{\sigma}^2_{e(\lambda_{GML})}$ and $\hat{\sigma}^2_{b(\lambda_{GML})}$. The respective GCV estimates are called $\hat{\sigma}^2_{e(\lambda_{GCV})}$ and $\hat{\sigma}^2_{b(\lambda_{GCV})}$, where

$$\hat{\sigma}^2_{e(\lambda_{GML})} = \frac{\mathrm{RSS}(\lambda_{GML})}{n-m}, \qquad \hat{\sigma}^2_{e(\lambda_{GCV})} = \frac{\mathrm{RSS}(\lambda_{GCV})}{n-m},$$
$$\hat{\sigma}^2_{b(\lambda_{GML})} = n\lambda_{GML}\hat{\sigma}^2_{e(\lambda_{GML})}, \quad \hat{\sigma}^2_{b(\lambda_{GCV})} = n\lambda_{GCV}\hat{\sigma}^2_{e(\lambda_{GCV})},$$

and $\mathrm{RSS}(\lambda_M)$ is the residual sum of squares when the estimated λ was computed using method M= GCV or M=GML.

We are reporting the Bias, $\mathrm{E}(\sigma_e) - \hat{\sigma}_{e(\lambda_M)}$, the standard error, SE=$\sqrt{\mathrm{Var}(\hat{\sigma}_{e(\lambda_M)})}$, and the Root Mean Squared Error, RMSE= $\sqrt{\mathrm{BIAS}^2 + \mathrm{SE}^2}$. The results of the simulation showed that the GCV estimate of σ_e has larger bias and variability than the GML estimate. The GCV estimate of the random-effects shows a better performance than the GML estimate when the variance of the errors and the random-effects are about the same. The GML estimate outperforms the GCV for case 2. These results are shown in Tables 1, 2 and 3 and plots with the distribution of the sampled variances are given in Figures 1 and 2. Figure 1 shows the box-plot comparison for the first case when $n = 18$ and Figure 2 shows the quantile plot for the sampled error variance simulated with $\sigma_e^2 = 16$ and $\sigma_b^2 = 24^2$. The theoretical quantiles are those of a Chi-squared random variable with 16 degrees of freedom.

Table 1: This table shows the simulation results for Case 1: $\sigma_e = 4$ and $\sigma_b = 24$.

			GCV			GML		
Case	**Reps**		**BIAS**	**SE**	**RMSE**	**BIAS**	**SE**	**RMSE**
1	**3**	$\boldsymbol{\sigma_e}$	0.09	7.2	7.2	-0.02	6.5	6.5
		$\boldsymbol{\sigma_b}$	1.2	413	413	-0.03	376	376
1	**10**	$\boldsymbol{\sigma_e^2}$	0.05	3.12	3.12	0.005	3.10	3.10
		$\boldsymbol{\sigma_b}$	-0.15	352	352	0.22	395	395

Table 2: This table shows the simulation results for Case 2: $\sigma_e = 24$ and $\sigma_b = 4$.

			GCV			GML		
Case	**Reps**		**BIAS**	**SE**	**RMSE**	**BIAS**	**SE**	**RMSE**
2	**3**	$\boldsymbol{\sigma_e^2}$	3.96	316	316.02	-0.96	200	200
		$\boldsymbol{\sigma_b}$	5.24	133	133.1	2.08	79	79.02
2	**10**	$\boldsymbol{\sigma_e^2}$	.94	120	120	-0.17	108	108
		$\boldsymbol{\sigma_b^2}$	1.01	38	38.01	-0.45	24	24

Table 3: This table shows the simulation results for Case 3: $\sigma_e = 5$ and $\sigma_b = 5$.

			GCV			GML		
Case	**Reps**		**BIAS**	**SE**	**RMSE**	**BIAS**	**SE**	**RMSE**
3	**3**	$\boldsymbol{\sigma_e^2}$	0.6	14	14.01	-0.02	11	11
		$\boldsymbol{\sigma_b}$	0.05	22	22	-0.5	18	18
3	**10**	$\boldsymbol{\sigma_e^2}$	0.08	5	5	-0.03	11	11
		$\boldsymbol{\sigma_b}$	0.07	18	18	-0.51	18	18

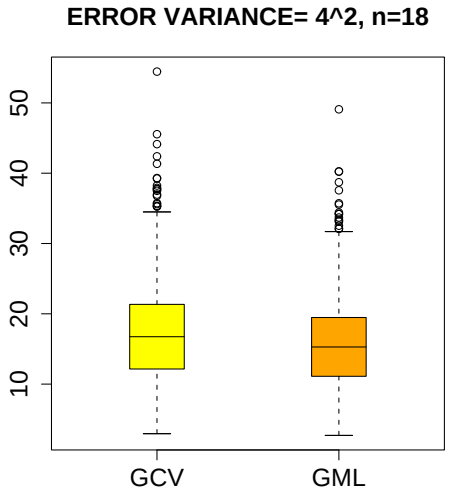

Figure 1: Box-plot of the distributions of 1000 simulated error variances from samples of size 18 and true error variance equal to 16.

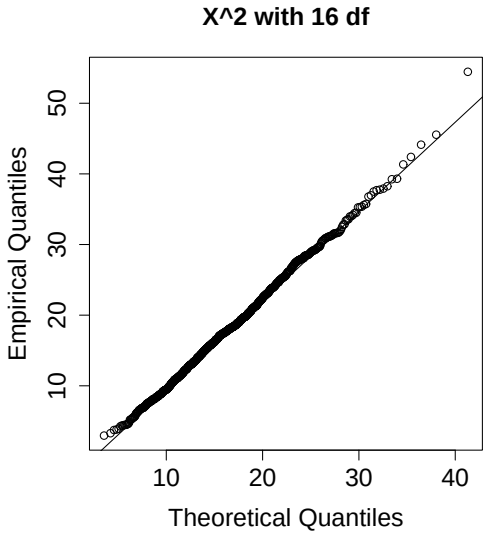

Figure 2: Plot of the theoretical quantiles of a Chi-squared random variable with 16 degrees of freedom vs. the empirical quantiles of the distribution of sampled error variances estimators for case 1 using the GCV criterion.

3.4 Summary

For several years now, it has been known that there is a relationship between the mixed-effects model, the cubic smoothing spline estimator and the conditional mean of the Bayesian model mentioned in Chapter II. We have extended those results to a general mixed-effects setting, a general penalized least squares error criterion and a general Bayesian model. We have also shown that the estimation of the variance components in the Bayesian or mixed-effects model, and/or the smoothing parameter in the penalized least squares framework, can be done via GML, GCV or UBR. We have examined the performance of the GCV estimator and compared it to the performance of the GML estimator. It seems that the GML estimator performs better than GCV though a larger simulation study, with a larger variety of scenarios is needed to provide adequate support for such a claim.

CHAPTER IV

KALMAN FILTERING

During the past years new developments in technology have made it easier to implement some statistical methods which earlier were too computationally intensive for use in practice. Increases in computer speed and memory capacity have alleviated storage and processing power problems to some degree. However, the same advances in technology are allowing us to gather much larger amounts of data. It is now common to see data sets with more than 200 predictors on 1000 subjects measured over long periods of time. The analysis of large data sets with many variables in the linear model framework will involve inversion of matrices producing flop counts that will be driven by a factor proportional to the cube of the number of variables in the model. For this reason, it is important to find efficient methods to compute our statistics in high dimensional settings. One popular computational method that can often be of great value in such cases is the Kalman Filter.

The Kalman filter was introduced in 1960 by Rudolph Kalman in the engineering literature as a recursive estimation procedure for a random model. The Kalman filter has been well known in the time series literature. But it was not until the 1980's that it started to attract attention from other areas of Statistics. The works of Ansley and Kohn (1985), Kohn and Ansley (1987, 1991), de Jong (1989) and Koopman and Durbin (1998) introduced the Kalman filter in the smoothing splines setting, whereas Sallas and Harville (1981) discussed the Kalman filter in the mixed-effetcs linear model context. More recently, Guo (2002, 2003) has promoted the use of the Kalman filter, in the smoothing spline setting, as an alternative to the use of mixed-effects procedures implemented in known software, such as SAS, when dealing with large

numbers of observations.

Here we will build our way up to the diffuse Kalman recursion through a series of simple steps. First, we will start by considering a purely random model and explain the forward (updating) and backward (smoothing) recursion algorithms for the Kalman filter in this context. The standard Kalman filter considers specific initial conditions that later will be changed to allow for a more general case, specifically, the use of diffuse conditions. While the diffuse Kalman filter is a little bit more complicated (e.g., see Ansley and Kohn, 1985; Kohn and Ansley, 1989; Koopman and Durbin, 1998), we will show a simple way to implement it using the approach taken by Eubank and Wang (2002) and Eubank, Huang and Wang (2003). We then illustrate its application in different linear model settings, among them linear models with correlated random errors and a functional linear model.

4.1 State-Space Models

Consider a signal-plus-noise model of the form

$$\boldsymbol{y}(t) \;=\; \boldsymbol{f}(t) + \boldsymbol{e}(t), \tag{4.1}$$

where $\boldsymbol{y}(t)$ is a $p \times 1$ vector of realizations from a continuous stochastic process observed at discrete points in time, $t = 1, \ldots, n$, $\boldsymbol{f}(t)$ is a vector corresponding to the signal part of the model evaluated at time t and the $\boldsymbol{e}(t)$ are vectors of unobservable, zero mean, normal random errors with a $p \times p$ variance-covariance matrix $W(t)$, and

$$\begin{aligned} W &= \{\mathrm{Cov}[\boldsymbol{e}(t), \boldsymbol{e}(s)]\}_{t,s=1,n} \\ &= \mathrm{diag}\{W(1), \ldots, , W(n)\}. \end{aligned} \tag{4.2}$$

The signal, $\boldsymbol{f}(t)$, is said to have a state-space representation if it can be written in the form

$$\boldsymbol{f}(t) \;=\; H(t)\boldsymbol{x}(t), \tag{4.3}$$

with

$$\boldsymbol{x}(t+1) \;=\; F(t)\boldsymbol{x}(t) + \boldsymbol{u}(t), \tag{4.4}$$

where $H(t)$ is a non random $p \times m$ matrix and $\boldsymbol{x}(t)$ is an $m \times 1$ random vector (called the state vector). The matrix $F(t)$ in (4.4) is a known $m \times m$ matrix called the transition matrix. The $\boldsymbol{u}(t)$ in (4.4) are $m \times 1$, normally distributed random vectors with zero means, positive-semidefinite variance-covariance matrix $R_u(t)$ and they are uncorrelated with each other and uncorrelated with the $\boldsymbol{e}(t)$. In this way,

$$\boldsymbol{y}(t) \;=\; H(t)\boldsymbol{x}(t) + \boldsymbol{e}(t) \tag{4.5}$$

is called the observation equation and (4.4) is called the state equation.

The initial goal here is to compute the predicted value of $\boldsymbol{f}(t)$, namely $\hat{\boldsymbol{f}}(t|t)$, given by $\mathrm{E}[\boldsymbol{f}(t)|\boldsymbol{y}(1), \ldots, , \boldsymbol{y}(t)]$. For this purpose, define

$$\boldsymbol{y}_t \;=\; [\boldsymbol{y}^T(1), \ldots, \boldsymbol{y}^T(t)]^T. \tag{4.6}$$

Then

$$\begin{aligned} \hat{\boldsymbol{f}}(t|t) &= \mathrm{E}[\boldsymbol{f}(t)|\boldsymbol{y}(1), \ldots, , \boldsymbol{y}(t)] \\ &= \mathrm{E}[\boldsymbol{f}(t)|\boldsymbol{y}_t]. \end{aligned} \tag{4.7}$$

But, (4.3) indicates that prediction of $\boldsymbol{f}(t)$ is tantamount to the prediction of $\boldsymbol{x}(t)$. Thus, let us define the BLUP of $\boldsymbol{x}(t)$ based on $\boldsymbol{y}(1), \ldots, , \boldsymbol{y}(t)$ as

$$\begin{aligned} \hat{\boldsymbol{x}}(t|t) &= \mathrm{E}[\boldsymbol{x}(t)|\boldsymbol{y}(1), \ldots, , \boldsymbol{y}(t)] \\ &= \mathrm{E}[\boldsymbol{x}(t)|\boldsymbol{y}_t]. \end{aligned} \tag{4.8}$$

The Kalman filter uses the special covariance structure that is a consequence of the recursive relationship given by (4.4) and (4.5) to provide an efficient, recursive

algorithm, that computes the predicted state vectors, and hence, the predicted signal vectors $\hat{\boldsymbol{f}}(t|t)$ and the smoothed prediction of $\boldsymbol{f}(t)$, namely

$$\hat{\boldsymbol{f}}(t) \quad = \quad \mathrm{E}[\boldsymbol{f}(t)|\boldsymbol{y}_n]. \tag{4.9}$$

These algorithms are called, respectively, the filtering (forward) and smoothing (backward) Kalman algorithms.

We will start by deriving the algorithms used by the standard Kalman filter. Equation (4.4) implies the need for specifying initial values for the state vector to start the recursion. The standard Kalman filter specifices that $\boldsymbol{x}(0) = 0$ and $\mathrm{Var}[\boldsymbol{x}(0)] = 0$. There exists some other ways to initialize the state vector $\boldsymbol{x}(0)$. Sallas and Harville (1981) chose $\boldsymbol{x}(0) = 0$ and suggested an approximate maximum likelihood procedure to estimate $\mathrm{Var}[\boldsymbol{x}(0)]$ whereas de Jong (1989) and Kohn and Ansley (1989) assumed $\boldsymbol{x}(0)$ to have a diffuse distribution. We will explain how to deal with diffuse initial conditions after deriving the Kalman filter for $\boldsymbol{x}(0) \equiv \mathbf{0}$.

First we will introduce some notation. Define the tth innovation as

$$\boldsymbol{\epsilon}(t) \quad = \quad \boldsymbol{y}(t) - E[\boldsymbol{y}(t)|\boldsymbol{y}(1), \ldots, \boldsymbol{y}(t-1)]. \tag{4.10}$$

This is the same as taking $\boldsymbol{\epsilon}(1) = \boldsymbol{y}(1)$ and then

$$\boldsymbol{\epsilon}(t) \quad = \quad \boldsymbol{y}(t) - \sum_{i=1}^{t-1} \mathrm{Cov}[\boldsymbol{y}(i), \boldsymbol{\epsilon}(i)] R_{\epsilon}^{-1}(i) \boldsymbol{\epsilon}(i), \tag{4.11}$$

for $t = 2, \ldots, n$ with

$$\mathrm{Var}[\boldsymbol{\epsilon}(t)] = R_{\epsilon}(t).$$

The set of innovation vectors, $\boldsymbol{\epsilon}(1), \ldots, \boldsymbol{\epsilon}(n)$, are uncorrelated and span the same vector space as the vectors of responses $\boldsymbol{y}(1), \ldots, \boldsymbol{y}(n)$. So, given any integer $j = 1, \ldots, n$, the computation of the BLUP of $\boldsymbol{x}(t)$ based on $\boldsymbol{y}(1), \ldots, \boldsymbol{y}(j)$, can also

be done using the innovations $\boldsymbol{\epsilon}(1), \ldots, \boldsymbol{\epsilon}(j)$. In this way, using multivariate analysis results, the BLUP of $\boldsymbol{x}(t)$ based on $\boldsymbol{y}(1), \ldots, \boldsymbol{y}(j)$ is found to be

$$\hat{\boldsymbol{x}}(t|j) = \sum_{i=1}^{j} \mathrm{Cov}[\boldsymbol{x}(t), \boldsymbol{\epsilon}(i)] R_{\epsilon}^{-1}(i) \boldsymbol{\epsilon}(i) \tag{4.12}$$

with respective prediction error variance-covariance matrix

$$S(t|j) = \mathrm{Var}[\boldsymbol{x}(t)] - \sum_{i=1}^{j} \mathrm{Cov}[\boldsymbol{x}(t), \boldsymbol{\epsilon}(i)] R_{\epsilon}^{-1}(i) \mathrm{Cov}[\boldsymbol{\epsilon}(i), \boldsymbol{x}(t)]. \tag{4.13}$$

Let $\hat{\boldsymbol{x}}(0|0) = 0$ and $S(0|0) = 0$. Then

$$\hat{\boldsymbol{x}}(t|t-1) = F(t-1)\hat{\boldsymbol{x}}(t-1|t-1), \tag{4.14}$$

with respective variance-covariance matrix

$$S(t|t-1) = F(t-1)S(t-1|t-1)F^T(t-1) + R_u(t-1). \tag{4.15}$$

Then, applying (4.5), (4.8), (4.10) and (4.12), we can now write

$$\boldsymbol{\epsilon}(t) = \boldsymbol{y}(t) - H^T(t)\hat{\boldsymbol{x}}(t|t-1) \tag{4.16}$$

and the variance-covariance matrix for $\boldsymbol{\epsilon}(t)$ takes the form

$$\mathrm{Var}[\boldsymbol{\epsilon}(t)] = H(t)S(t|t-1)H^T(t) + W(t) \tag{4.17}$$

$$= R_{\epsilon}(t). \tag{4.18}$$

From the distribution of

$$[\boldsymbol{x}(t), \boldsymbol{\epsilon}(t)]^T \mid \boldsymbol{y}(1) \ldots, \boldsymbol{y}(t-1)$$

it can then be shown that

$$\hat{\boldsymbol{x}}(t|t) = \hat{\boldsymbol{x}}(t|t-1) + S(t|t-1)R_{\epsilon}^{-1}(t)H(t)\boldsymbol{\epsilon}(t) \tag{4.19}$$

with

$$S(t|t) = S(t|t-1) - S(t|t-1)H^T(t)R_{\epsilon}^{-1}(t)H(t)S(t|t-1). \tag{4.20}$$

All this can be summarized as follows:

Algorithm 4.1.1 (Forward Recursion) *Initialize the forward recursion with* $\hat{\boldsymbol{x}}(0|0) = 0$ *and* $S(0|0) = 0$. *Then, for* $t = 1, \ldots, n$ *compute*

$$\begin{aligned}
S(t|t-1) &= F(t-1)S(t-1|t-1)F^T(t-1) + R_u(t-1), \\
R_\epsilon(t) &= H(t)S(t-1|t-1)H^T(t) + W(t), \\
\boldsymbol{\epsilon}(t) &= \boldsymbol{y}(t) - H(t)F(t-1)\hat{\boldsymbol{x}}(t-1|t-1), \\
\hat{\boldsymbol{x}}(t|t) &= F(t-1)\hat{\boldsymbol{x}}(t-1|t-1) + S(t|t-1)H^T(t)R_\epsilon^{-1}(t)\boldsymbol{\epsilon}(t), \\
K(t) &= F(t)S(t|t-1)H^T(t)R_\epsilon^{-1}(t), \\
S(t|t) &= S(t|t-1) - S(t|t-1)H^T(t)R_\epsilon^{-1}(t)H(t)S(t|t-1).
\end{aligned}$$

At the end of algorithm (4.1.1) we have computed

$$\mathrm{E}[\boldsymbol{f}(1)|\boldsymbol{y}_1], \mathrm{E}[\boldsymbol{f}(2)|\boldsymbol{y}_2], \ldots, \mathrm{E}[\boldsymbol{f}(n)|\boldsymbol{y}_n].$$

But often times, what we want is

$$\mathrm{E}[\boldsymbol{f}(1)|\boldsymbol{y}_n], \mathrm{E}[\boldsymbol{f}(2)|\boldsymbol{y}_n], \ldots, \mathrm{E}[\boldsymbol{f}(n)|\boldsymbol{y}_n],$$

and the backward recursion of the Kalman filter allows us to compute these quantities.

First, notice that

$$\hat{\boldsymbol{f}}(t) = \boldsymbol{y}(t) - \mathrm{E}[\boldsymbol{e}(t)|\boldsymbol{y}_n] \tag{4.21}$$

Letting $\hat{\boldsymbol{e}}(t) = \mathrm{E}[\boldsymbol{e}(t)|\boldsymbol{y}_n]$, we then have

$$\hat{\boldsymbol{e}}(t) = \mathrm{Cov}[\boldsymbol{e}(t), \boldsymbol{y}_n][\mathrm{Var}(\boldsymbol{y}_n)]^{-1}\boldsymbol{y}_n \tag{4.22}$$

and this is also equivalent to

$$\hat{\boldsymbol{e}}(t) = \mathrm{E}[\boldsymbol{e}(t)|\boldsymbol{\epsilon}_n] \tag{4.23}$$

with $\boldsymbol{\epsilon}_n = [\boldsymbol{\epsilon}(1), \ldots, \boldsymbol{\epsilon}(n)]^T$. Then, the backward recursion works as follows:

Algorithm 4.1.2 (Backward Recursion) *Pass on the values $K(t)$, $\boldsymbol{\epsilon}(t)$ and $R_\epsilon(t)$ from the forward recursion and set*

$$\begin{aligned} \boldsymbol{d}(n) &= \boldsymbol{\epsilon}(n), \\ C(n) &= R_\epsilon(n), \\ R_\epsilon(n) &= I - R_\epsilon^{-1}(n), \end{aligned}$$

and

$$\boldsymbol{\epsilon}(n) = \boldsymbol{\epsilon}(n)C^{-1}(n).$$

Then, for $t = (n-1), \ldots, 1$ compute

$$\begin{aligned} \hat{\boldsymbol{x}}(t|n) &= -[H^T(t)C^{-1}(t)\boldsymbol{d}^T(t)] + [F(t+1) \\ &\quad -K(t+1)H^T(t)]^T\hat{\boldsymbol{x}}(t+1|n), \\ S(t|n) &= H^T(t)C^{-1}(t)H(t) \\ &\quad +[F(t+1) - K(t+1)H(t)]^T S(t|t+1) \\ &\quad \times[F(t+1) - K(t+1)H(t)], \\ \boldsymbol{d}(t) &= \boldsymbol{\epsilon}(t), \\ C(t) &= R_\epsilon(t), \\ \hat{\boldsymbol{e}}(t) &= \epsilon(t)C^{-1}(t) + K^T(t)\hat{\boldsymbol{x}}(t|n), \\ R_\epsilon(t) &= I - R_\epsilon^{-1}(t) - K^T(t)S(t|n)K(t). \end{aligned}$$

Eubank and Wang (2002) showed that the Kalman filter is equivalent to a Cholesky algorithm. Generally, the Cholesky decomposition requires computations of order n^2, unless the variance-covariance matrix $\mathrm{Var}[\boldsymbol{y}_n]$, of the system of equations

$$\mathrm{Var}[\boldsymbol{y}_n]\boldsymbol{b} = \boldsymbol{c}, \tag{4.24}$$

possesses some type of structure, i.e., band limited. Eubank and Wang (2002) illustrated that the state-space structure in model (4.1) allows for the development of a smart Cholesky decomposition by writing

$$\text{Var}[\boldsymbol{y}_n] \;=\; LR_\epsilon L^T \tag{4.25}$$

with $R_\epsilon = \text{diag}\{R_\epsilon(t)\}_{t=1,n}$, and L a lower triangular matrix. Then

$$\boldsymbol{\epsilon}_n \;=\; L^{-1}\boldsymbol{y}_n. \tag{4.26}$$

Now, recall that

$$\hat{\boldsymbol{f}}_n \;=\; \text{Cov}(\boldsymbol{f}, \boldsymbol{y}_n)[\text{Var}(\boldsymbol{y}_n)]^{-1}\boldsymbol{y}_n, \tag{4.27}$$

for $\boldsymbol{f} = [\boldsymbol{f}^T(1), \ldots, \boldsymbol{f}^T(n)]^T$. Hence, using (4.25) we can write

$$\hat{\boldsymbol{f}}_n \;=\; \boldsymbol{y}_n - W(L^T)^{-1}R_\epsilon^{-1}\boldsymbol{\epsilon}_n. \tag{4.28}$$

But, by (4.22), we know that

$$\begin{aligned} \hat{\boldsymbol{e}} \;&=\; W(L^T)^{-1}R_\epsilon^{-1}L^{-1}\boldsymbol{y}_n \\ &=\; W(L^T)^{-1}R_\epsilon^{-1}\boldsymbol{\epsilon}_n, \end{aligned} \tag{4.29}$$

which is obtained after the backward Kalman filter algorithm.

Now, the question arises of what to do when it is not advisable to choose the initial state vector to be zero. For example, there are some cases, e.g., the integrated Brownian motion model, where the initial state vector is zero. But, this is not usually the case. We may want to model the signal $\boldsymbol{f}(\cdot)$ with some mean different from zero but we have no idea of what values the state vector $\boldsymbol{x}(0)$ can take on. In this case, it is suggested to model the second moment of $\boldsymbol{x}(0)$ with a diffuse prior (see Kohn and Ansley, 1989; Koopman and Durbin, 1998).

Without loss of generality, suppose that $\boldsymbol{x}(0|0) = \boldsymbol{\mu}_{\boldsymbol{x}}$, and that $\text{Var}[\boldsymbol{x}(0)] = \nu I$. The BLUP $\hat{\boldsymbol{x}}(t-1|t-1)$, and its respective variance-covariance matrix $S(t-1|t-1)$, are expressed in terms of conditional expectations (see (4.8)). In analogous ways as the ones used to derived the conditional mean and variance of the Bayesian model in Chapter II, it can be shown that the limits of the BLUP $\hat{\boldsymbol{x}}(t-1|t-1)$, its variance, the innovations, etc. do not depend on the parameter ν and can be calculated but not with the standard Kalman filter.

The problem is that our signal vector, $\boldsymbol{f}$, now has the form

$$\boldsymbol{f} \;=\; T\boldsymbol{\theta} + U\boldsymbol{\gamma}$$

with $T\boldsymbol{\theta}$ representing the mean and $U\boldsymbol{\gamma}$ the random part of the model and it is precisely this type of model that we are dealing with here. The standard Kalman filter only shows us how to deal with the random term $U\boldsymbol{\gamma}$.

There exists several algorithms to deal with diffuse initial conditions like the ones proposed by Ansley and Kohn (1985), Kohn and Ansley (1989) and Koopman and Durbin (1998). Some of these algorithms need additional multiplications, etc., in both, the forward and backward recursions, that make them more complicated that the simple recursions for the standard case. Eubank, Huang and Wang (2003) developed a way to circumvent such problems. We will illustrate their approach using as an example the Bayesian model (2.64).

Consider observations $y(t_i)$ of the form

$$y(t_i) \;=\; f(t_i) + e(t_i) \tag{4.30}$$

for $i = 1, \ldots, n$ and with $f(t_i) = \sigma_b X(t_i)$, where $X(\cdot)$ is the stochastic process

$$X(t) \;=\; \int_0^1 \frac{(t-u)_+^{m-1}}{(m-1)!} dW(u), \tag{4.31}$$

with

$$(t)_+ = \begin{cases} 0 & \text{if} \quad t < 0 \\ t & \text{if} \quad t \geq 0 \end{cases},$$

and $W(\cdot)$ a Wiener process: i.e., a zero mean normal process with stationary independent increments and $W(0) = 0$. Here we define $\int_0^1 g(u)dW(u)$ as the limit of the Riemann-Stieljes sum $\sum_P g(u_i)[W(u_{i+1}) - W(u_i)]$ with P a partition of $[0, 1]$ and assume that the Wiener process $W(\cdot)$ is independent of $e(t_1), \ldots, e(t_n)$. The errors $e(t_i)$ are independently, normally distributed random variables with zero mean and variance σ_e^2.

Notice that,

$$\text{Cov}[X(t), X(s)] = \text{E}\left[\int_0^1 \frac{(t-u)_+^{m-1}}{(m-1)!}dW(u) \int_0^1 \frac{(s-u)_+^{m-1}}{(m-1)!}dW(u)\right].$$

Using the independent increments property and the Riemann-Stieljes integration we have

$$\text{Cov}[X(t), X(s)] = \int_0^1 \frac{(t-u)_+^{m-1}}{(m-1)!}\frac{(s-u)_+^{m-1}}{(m-1)!}du.$$

We are now going to show that $f(t_i) = \sigma_b X(t_i)$ has state-space representation. By definition (4.31) we can write

$$\begin{aligned} X(t_{i+1}) &= \int_0^{t_{i+1}} \frac{(t_{i+1}-u)^{m-1}}{(m-1)!}dW(u) \\ &= \int_0^{t_i} \frac{(t_{i+1}-u)^{m-1}}{(m-1)!}dW(u) + \int_{t_i}^{t_{i+1}} \frac{(t_{i+1}-u)^{m-1}}{(m-1)!}dW(u) \end{aligned} \quad (4.32)$$

Let

$$u(t_i) = \int_{t_i}^{t_{i+1}} \frac{(t_{i+1}-u)^{m-1}}{(m-1)!}dW(u) \quad (4.33)$$

so that for $t_i < t_j$

$$\begin{aligned} \text{Cov}[u(t_i), u(t_j)] &= \int_{t_i}^{t_j} \frac{(t_i-u)^{m-1}(t_j-u)^{m-1}}{[(m-1)!]^2}du \\ &= R_{ij}. \end{aligned}$$

Now we need to work with

$$\int_0^{t_i} \frac{(t_{i+1}-u)^{m-1}}{(m-1)!} dW(u).$$

Adding and subtracting t_i in the numerator, we obtain

$$\int_0^{t_i} \frac{(t_{i+1}-u)^{m-1}}{(m-1)!} dW(u) = \int_0^{t_i} \frac{[(t_{i+1}-t_i)+(t_i-u)]^{m-1}}{(m-1)!} dW(u).$$

Applying the binomial theorem and interchanging the sum and the integration we get

$$\begin{aligned} \int_0^{t_i} \frac{(t_{i+1}-u)^{m-1}}{(m-1)!} dW(u) &= \sum_{k=0}^{m-1} \frac{(t_{i+1}-t_i)^k}{k!} \int_0^{t_i} \frac{(t_i-u)^{m-k-1}}{(m-k-1)!} dW(u), \\ &= \sum_{k=0}^{m-1} \frac{(t_{i+1}-t_i)^k}{k!} X^{(k)}(t_i). \end{aligned} \tag{4.34}$$

In this way, using (4.33) and (4.34) we can rewrite $X(t_{i+1})$ as

$$X(t_{i+1}) = \sum_{k=0}^{m-1} \frac{(t_{i+1}-t_i)^k}{k!} X^{(k)}(t_i) + u(t_i). \tag{4.35}$$

Thus, define the state vector as $\boldsymbol{X}(t_i) = [X(t_i), X^{(1)}(t_i), \ldots, X^{(m-1)}(t_i)]^T$. Then, we can take

$$F(t_i) = \begin{pmatrix} 1 & (t_{i+1}-t_i) & \frac{(t_{i+1}-t_i)^2}{2!} & \ldots & \frac{(t_{i+1}-t_i)^{m-1}}{(m-1)!} \\ 0 & 1 & (t_{i+1}-t_i) & \ldots & \frac{(t_{i+1}-t_i)^{m-2}}{(m-2)!} \\ . & 0 & 1 & \ldots & \frac{(t_{i+1}-t_i)^{m-3}}{(m-3)!} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & . & . & . & 1 \end{pmatrix} \tag{4.36}$$

and using $\boldsymbol{X}(t_i)$ and (4.36) in model (4.30), we have

$$y(t_i) = \mathbf{h}^T \boldsymbol{X}(t_i) + e(t_i), \tag{4.37}$$

and

$$\boldsymbol{X}(t_{i+1}) = F(t_i)\boldsymbol{X}(t_i) + \mathbf{u}(t_i) \tag{4.38}$$

with $\mathbf{h}^T = (1, 0, \ldots, 0)^T$ an $m \times 1$ vector. Together, equations (4.37) and (4.38) define a simple state-space model with the same covariance structure that we have been managing in the Bayesian model with the polynomial trend, i.e.,

$$\text{Var}[\boldsymbol{y}] \;=\; \sigma_e^2 Q,$$

for $\boldsymbol{y} = [y(t_1), \ldots, y(t_n)]^T$ and

$$Q \;=\; (n\lambda R + I),$$

where

$$R \;=\; \{R_{ij}\}_{i,j=1,n}\,.$$

This simple state-space model has the advantage of not needing an improper prior so we can use the standard Kalman filter to compute predictors of the state vector and the signal.

As we saw in (4.22), the implementation of the standard Kalman filter on the vector $\boldsymbol{y}$ yields $Q^{-1}\boldsymbol{y}$ (since $W(t_i) = \sigma_e^2$ for all i). We have shown that, in this setting, the BLUP $\hat{\boldsymbol{f}}$ is given by

$$[I - Q^{-1}(I - T(T^T Q^{-1} T)^{-1} T^T Q^{-1})]\boldsymbol{y}.$$

Now, if we apply the Kalman filter to $\boldsymbol{y}$ and each of the m columns of T, we obtain: $Q^{-1}T$ and $Q^{-1}\boldsymbol{y}$. So, if we set $C = Q^{-1}T$ and $\mathbf{z} = Q^{-1}\boldsymbol{y}$, the computation of $\hat{\boldsymbol{f}}$ reduces to computing $T(T^TC)^{-1}T^T\mathbf{z}$. This can be done by solving the system of equations $T^TCB = T^T\mathbf{z}$, for some matrix B, and this can be accomplished in order m operations since T^TC is an $m \times m$ matrix and $T^T\mathbf{z}$ is an $m \times 1$ vector.

Usually, as we saw in previous chapters, we are required to estimate not only $\boldsymbol{f}$ but also the smoothing parameter and/or the variance components of the random

terms. The sample likelihood can be obtained after the forward Kalman recursion has been completed. De Jong (1988) showed that the likelihood for a state space model can be evaluated using the Kalman filter with initial conditions $\hat{\boldsymbol{x}}(0|0) = 0$ and $S(0|0) = 0$ and established that, apart from a constant

$$-2\ell[\boldsymbol{y}(1)\ldots,\boldsymbol{y}(n)] \;=\; \sum_{t=1}^{n}\log|R_\epsilon(t)| + \sum_{t=1}^{n}\boldsymbol{\epsilon}^T(t)R_\epsilon^{-1}(t)\boldsymbol{\epsilon}(t). \tag{4.39}$$

The GCV and UBR criteria can also be used to estimate the smoothing parameter and the variance components by computing the diagonal elements of the matrix $(I - A_\lambda)$ after the backward recursion (see Eubank et al., 2003). Another advantage of the Kalman filter is that it allow us to compute the predictors of the derivatives of $\boldsymbol{f}(t)$. Notice that if we interchange the 1 and the firs zero in the vector $\boldsymbol{h}$ in our simple model, what we are obtaining is the firs derivative of f. In this way, we can obtain estimates of any of the $(m-1)$ derivatives of the signal with the same computational effort.

4.2 Examples

One of the goals of this dissertation is to show how to employ the Kalman filter for different types of scenarios. In the existing literature, when working with smoothing spline estimators, it has been shown how to obtain the estimator using, e.g., the SAS procedure PROC MIXED. But, when suggesting the use of the Kalman filter no code has been shown. In this section, we will explicitly illustrate the different vectors and matrices for two different settings: a signal-plus-noise model whose errors are modeled with an autoregressive process of order 1, AR(1), and a functional linear model. The codes to compute the corresponding estimators are provided in Appendix A and B.

4.2.1 Models with Correlated Random Errors

Our first example involves models with correlated random errors. Along the development of this dissertation, we have assumed that the errors $e(t_i)$ are uncorrelated with the same error variance σ_e^2. But often times, it is required to assume some other type of covariance structure. We will show that, as long as the covariance structure of the errors can be written with a state-space representation, we can still apply the Kalman filter algorithm. To illustrate this case, we use an example taken from Wang (1998a).

Wang used a data set taken from the Box and Jenkins's book "Time Series Analysis" (1976). The data set consists of 197 measurements of the "uncontrolled" concentration in a continuous chemical process sampled every two hours (see Figure 3).

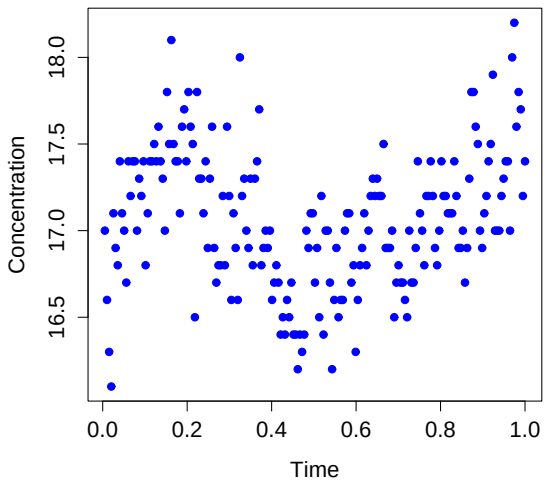

Figure 3: Time Series A from Box and Jenkin's book (1976). The data consists of 197 measurements of the "uncontrolled" concentration in a continuous chemical process sampled every two hours.

We will now illustrate how to compute the cubic smoothing spline estimator for this series using the Kalman filter. For this purpose, first note that we can represent our function as

$$\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{e} \tag{4.40}$$

with $\boldsymbol{y} = [y(t_1), \ldots, y(t_n)]$ the vector of responses and $\boldsymbol{f} = [f(t_1), \ldots, f(t_n)]$ modeled as

$$\boldsymbol{f} = T\boldsymbol{\theta} + R\boldsymbol{b}, \tag{4.41}$$

where T, $\boldsymbol{\theta}$, $\boldsymbol{b}$ and R are as is in (2.41)-(2.44). The vector of random errors, $\boldsymbol{e}$, is normally distributed with zero mean and variance-covariance matrix given by

$$\text{Var}[\boldsymbol{e}] = W \tag{4.42}$$

with W depending on some correlation parameter ρ and variance σ_e^2 but otherwise having a known structure.

Then, the smoothing spline estimator of $\boldsymbol{f}$ will be the minimizer of

$$(\boldsymbol{y} - T\boldsymbol{\theta} - R\boldsymbol{b})^T W^{-1} (\boldsymbol{y} - T\boldsymbol{\theta} - R\boldsymbol{b}) + \frac{1}{n\lambda} \boldsymbol{b}^T R \boldsymbol{b}.$$

Similarly to the steps taken to obtain equations (2.51) and (2.52), we can find $\boldsymbol{\theta}$ and $\boldsymbol{b}$ by solving the system of equations:

$$T^T W^{-1} T\boldsymbol{\theta} = T^T W^{-1} (\boldsymbol{y} - \boldsymbol{b}) \tag{4.43}$$

$$Q_W \boldsymbol{b} = n\lambda (\boldsymbol{y} - T\boldsymbol{\theta}), \tag{4.44}$$

with

$$Q_W = n\lambda R + W. \tag{4.45}$$

Solving for $\boldsymbol{\theta}$ and $\boldsymbol{b}$, it can be proved that the estimator $\hat{\boldsymbol{f}}$ will then have the form:

$$[I - Q_W^{-1}(I - T(T^T Q_W^{-1} T)^{-1} T^T Q_W^{-1})]\boldsymbol{y}.$$

To apply the standard Kalman filter to the columns of T and to $\boldsymbol{y}$ we need to write $y(t_i)$ using a state-space representation. For this, we will use again our simple model. We know that the numerical solution of the smoothing spline estimator is the same as the one for the Bayesian model

$$\boldsymbol{y} \;=\; T\boldsymbol{\theta} + \sigma_b \boldsymbol{X} + \boldsymbol{e},$$

with assumptions as in (2.64). But we have shown previously that, for the simple model $\sigma_b \boldsymbol{X} + \boldsymbol{e}$ with independent errors $\boldsymbol{e}$, we can find a state-space representation. So, we need to show that we can express $\sigma_b \boldsymbol{X} + \boldsymbol{e}$ as a state-space model with the $\boldsymbol{e}$ being generated by an AR(1).

We will proceed as follows: since the errors $e(t_i)$ are generated by an AR(1) process, they can be written as

$$e(t_{i+1}) \;=\; be(t_i) + v(t_i), \tag{4.46}$$

with b a non random coefficient and the $v(t_i)$ independently, normally distributed random errors with zero mean and variance σ_e^2. Then, we can write $y(t_i)$ as

$$y(t_i) \;=\; \boldsymbol{h}_*^T(t_i)\boldsymbol{X}_*(t_i), \tag{4.47}$$

$$\boldsymbol{X}_*(t_{i+1}) \;=\; F_*(t_i)\boldsymbol{X}_*(t_i) + \boldsymbol{u}_*(t_i), \tag{4.48}$$

with

$$\boldsymbol{h}_*^T \;=\; [\boldsymbol{h}^T(t_i), 1]^T, \tag{4.49}$$

and

$$\mathbf{X}_*(t_i) \;=\; [\boldsymbol{X}^T(t_i), e(t_i)]^T. \tag{4.50}$$

The errors are represented by

$$\boldsymbol{u}_*(t_i) \quad = \quad [\boldsymbol{u}(t_i), v(t_i)], \tag{4.51}$$

with $\boldsymbol{h}(t_i) = (1,0)^T$, and $\boldsymbol{X}(t_i)$ and $\boldsymbol{u}(t_i)$ as in (4.31) and (4.33), respectively. The matrix F_* takes the form

$$\begin{pmatrix} F(t_i) & \mathbf{0} \\ \mathbf{0} & b \end{pmatrix}, \tag{4.52}$$

with $F(t_i)$ defined as in (4.36).

The resulting smoothing spline estimator is shown in Figure 4. We estimated the correlation coefficient b using the GML method and found it to be = 0.99999. The variance was estimated to be 0.0052 and $\hat{\lambda}_{\text{GML}} = 0.0000552$.

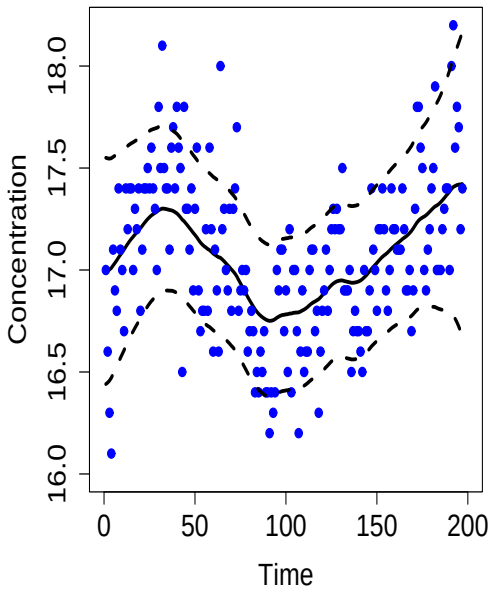

Figure 4: Smoothing spline estimator of the time series A from Box and Jenkin's book (1976) using the Kalman filter. The parameters were estimated using the GML criterion and were found to be: $\hat{b} = 0.99999$, $\hat{\sigma}_e^2 = 0.0052$ and $\hat{\lambda} = 0.0000552$.

4.2.2 A Functional Linear Model for Nested and Crossed Samples of Curves

The purpose of this example is twofold: first, we want to exemplify the use of the Kalman filter in the context of functional data analysis. Secondly, we want to give the reader an idea of the computational advantages of applying the Kalman filter versus other approaches. To accomplish these objectives we have chosen to compare our method with the approach taken by Brumback and Rice (1998) in their article "Smoothing Spline Models for the Analysis of Nested and Crossed Samples of Curves".

In their article, the goal was to analyze metabolite progesterone profileus, measured daily in urine over the course of a menstrual cycle in a group of 51 women. The study is part of a continuing study of early pregnancy loss carried out by the Institute for Toxicology and Environmental Health at the University of California, Davis, in collaboration with the Reproductive Epidemiology Section of the California Department of Health Services, Berkeley.

The sample comes from patients with healthy reproductive function participating in an artificial insemination clinic. The women in the study were divided into two groups: 29 in the non-conceptive group and 22 in the conceptive group. Each woman contributed a different number of cycles, ranging from 1 to 5 cycles, and some of the cycles have missing values. Figure 5 shows profiles for the same women corresponding to different cycles.

For illustration purposes, we will adopt the model proposed by Brumback and Rice: i.e., we consider functions of the form

$$y_{ijk}(t) \quad = \quad f_i(t) + f_{i,j}(t) + f_{ijk}(t) + e_{ijk}(t), \tag{4.53}$$

with $i = 1, 2$, $j = 1, \ldots, , n_i$, $k = 1, \ldots, n_{ij}$ and $t = -8, -7, \ldots, 15$. Here, $y_{ijk}(t)$ is the hormone measurement observed at time t from the kth cycle belonging to the jth

woman in the ith group.

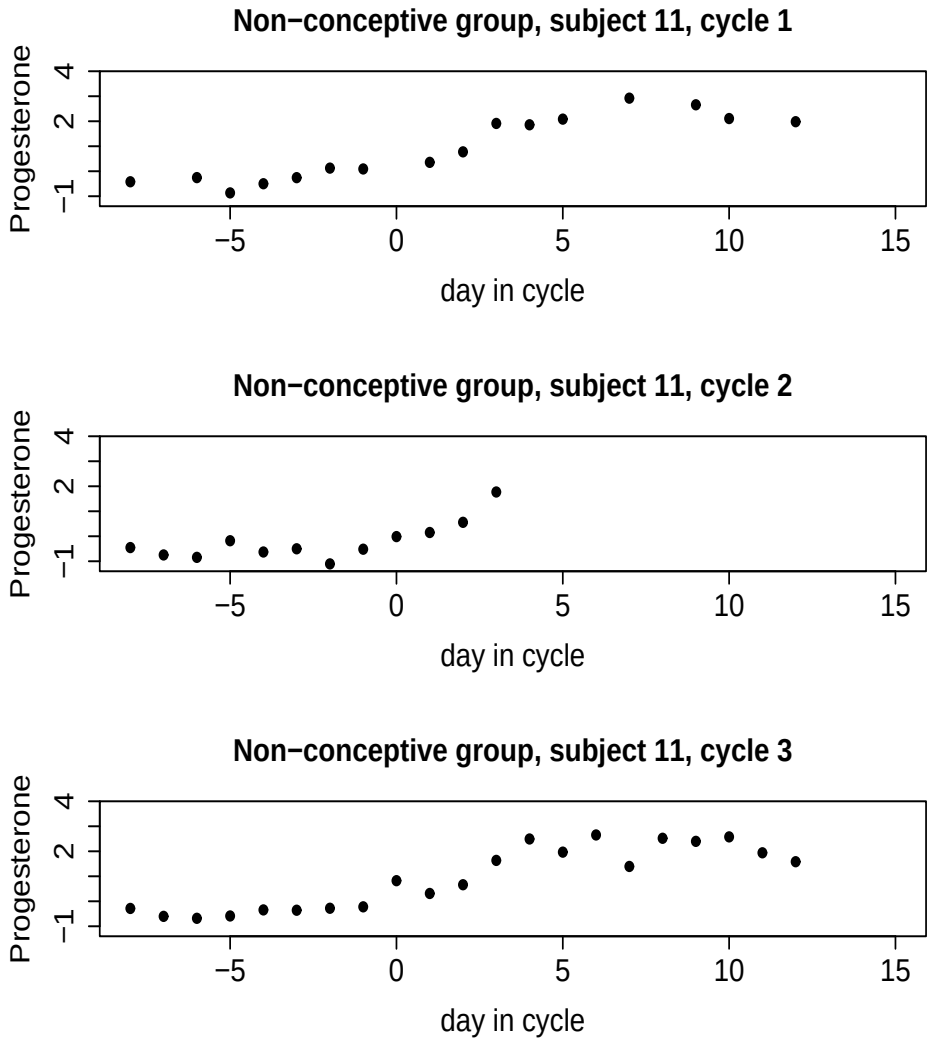

Figure 5: Observed progesterone measurements for subject 11 in the non-conceptive group. The plots correspond to three of the four cycles for subject 11 and show the log concentration versus day in the cycle. All cycles have missing observations. Days corresponding to the menses were excluded.

The functions $f_i(\cdot)$, $f_{ij}(\cdot)$ and $f_{ijk}(\cdot)$, represent smooth functions corresponding to a group mean, subject departure from the group mean, and cycle departure from

the subject mean, respectively. The errors $e_{ijk}(t)$ are iid normal random variables with zero mean and variance σ_e^2. The errors do not depend on t. The notation $e_{ijk}(t)$ is meant to indicate that it is the unexplained variability corresponding to the measurement taken at time t for individual j, cycle k and group i.

In their paper, Brumback et al. made use of the relationship between smoothing spline estimators and mixed-effects models to write an individual profile by its vector representation

$$\boldsymbol{y}_{ijk} = T^{ijk}\boldsymbol{\theta}_i + U^{ijk}\boldsymbol{\gamma}_i + T^{ijk}\boldsymbol{\theta}_{ij} + U^{ijk}\boldsymbol{\gamma}_{ij} + T^{ijk}\boldsymbol{\theta}_{ijk} + U^{ijk}\boldsymbol{\gamma}_{ijk} + \boldsymbol{e}_{ijk}, \quad (4.54)$$

with $\boldsymbol{y}_{ijk} = [y_{ijk}(-8), \ldots, y_{ijk}(15)]^T$, $T^{ijk}\boldsymbol{\theta}_i + U^{ijk}\boldsymbol{\gamma}_i$ the mixed-effects model representation of the group mean function, $T^{ijk}\boldsymbol{\theta}_{ij} + U^{ijk}\boldsymbol{\gamma}_{ij}$ the mixed-effects model representation of the subject departure from the group mean, and $T^{ijk}\boldsymbol{\theta}_{ijk} + U^{ijk}\boldsymbol{\gamma}_{ijk}$ the mixed-effects model representation of the cycle departure from the subject mean. The design matrices T^{ijk} and U^{ijk} indicate the time points specific to curve ijk. Once having written the mixed-effects model representation for each individual curve, they took the ordered set of unique observation times from all curves taken together, and then they defined the terms of each of the mixed-effects models in (4.54) in the same way Wang (1998b) defined his in model 3 (see Chapter II, pp. 31): e.g., R^{ijk} is defined as in (2.44) and having the decomposition $R^{ijk} = Z^{ijk}(Z^{ijk})^T$ and the $U^{ijk} = Z^{ijk}$.

Setting the random effects of curves in different grouping factor to be independent of each other (i.e., $\boldsymbol{\gamma}^1$ is independent of $\boldsymbol{\gamma}^2$, $\boldsymbol{\gamma}^{ij}$ is independent of $\boldsymbol{\gamma}^{ij'}$, for $j \neq j'$ and of the $\boldsymbol{\gamma}^i$, etc.), and of the random errors $\boldsymbol{e}_{ijk}$, for all i, j and k, they stacked the curves in an analogous way to the univariate case (1.11), resulting in a big ANOVA model

$$\boldsymbol{y} = T^g\boldsymbol{\theta}_g + U^g\boldsymbol{\gamma}_g + T^s\boldsymbol{\theta}_s + U^s\boldsymbol{\gamma}_s + T^c\boldsymbol{\theta}_c + U^c\boldsymbol{\gamma}_c + \boldsymbol{e}, \quad (4.55)$$

with $\boldsymbol{y} = [\boldsymbol{y}_{1,1,1}, \ldots, \boldsymbol{y}_{2,29,1}]$, $T^g = \text{diag}\{T^{ijk}\}$, etc.

After writing (4.55), they wrote its equivalent penalized least squares representation

$$\begin{aligned}&\text{argmin}\sum_i\sum_k\sum_k\sum_t[y_{ijk}(t)-f_i(t)-f_{i,j}(t)-f_{ijk}(t)]^2\\&+\lambda_g\int[f_i^{(2)}(t)]^2dt+\lambda_s\int[f_{ij}^{(2)}(t)]^2dt+\lambda_c\int[f_{ijk}^{(2)}(t)]^2dt.\end{aligned}$$

Brumback and Rice pointed out that computation of the smoothing spline estimator can then be done using standard software, like SAS, using the mixed-effects model framework. But, since this data set consists of 2183 observations, already implemented routines will take a massive amount of time trying to invert the matrices required. So, they proposed a transformation of the data.

Notice that (4.55) can be reduce even further as

$$\boldsymbol{y} = T\boldsymbol{\theta}+U\boldsymbol{\gamma}+\boldsymbol{e} \tag{4.56}$$

where

$$\begin{aligned}T &= [T^g,T^s,T^c],\\ \boldsymbol{\theta} &= [\boldsymbol{\theta}_g,\boldsymbol{\theta}_s,\boldsymbol{\theta}_c]^T,\\ U &= [U^g,U^s,U^c]\end{aligned}$$

and

$$\boldsymbol{\gamma} = [\boldsymbol{\gamma}_c,\boldsymbol{\gamma}_c,\boldsymbol{\gamma}_c]^T.$$

Based on this fact, they proposed to use a transformation matrix whose rows are the orthogonal eigenvectors of TT^T corresponding to the zero eigenvalues. Then they applied the EM algorithm of Dempster, Laird and Rubin (1977) to maximize the restricted likelihood in the variance components and thus, obtaining estimators of the smoothing parameters λ_g, λ_s and λ_c which they assumed were equal. Estimation

of the function corresponding to the conceptive and non-conceptive groups took approximately one hour and twenty minutes, including the numerical optimization for the smoothing parameters. Figure 6 shows the smoothing spline estimators for both, conceptive and non conceptive groups.

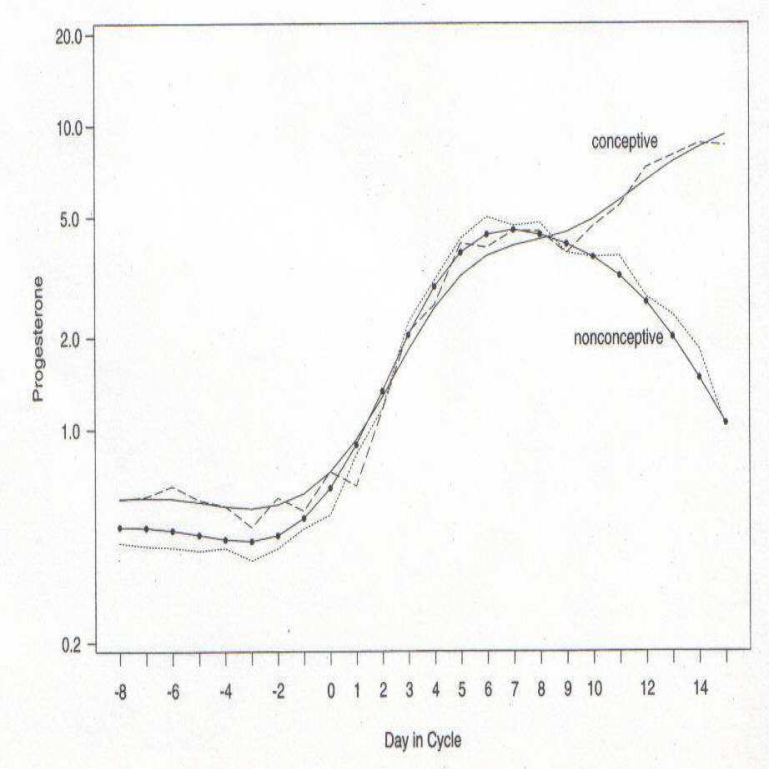

Figure 6: Sample of urinary metabolite progesterone curves measured over 21 conceptive and 70 non conceptive menstrual cycles. Smooth estimates for the non conceptive and conceptive group means obtained by Brumback and Rice. The picture was scanned from the Brumback and Rice article published by JASA (1998).

Using their model formulation, we applied the Kalman filter to obtain the smoothing spline estimators for both group functions. We separated the observation in their respective groups and ordered them according to time t. Then, we ran the forward

pass of the Kalman filter to adaptively select the smoothing parameters. Once the smoothing parameters were selected, we ran the Kalman filter to obtain the smoothing spline estimators. Our fitted functions (Figure 7) give a pretty close visual agreement to those in Brumback and Rice (1998) work.

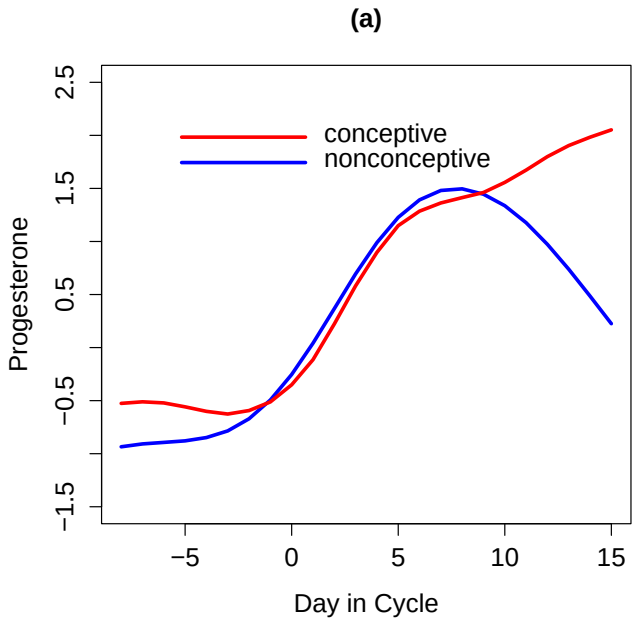

Figure 7: Sample of urinary metabolite progesterone curves measured over 21 conceptive and 70 non conceptive menstrual cycles. Smooth estimates obtained using the Kalman filter.

To get an idea of the variability of the mean group functions, Brumback and Rice proposed a bootstrap algorithm using a hierarchical sampling scheme. To obtain the bootstrap sample they did the following

1. Draw a sample of size g with replacement from the total number of women in group i.

2. For the jth sampled woman, $j = 1, \ldots, g$, draw a sample with replacement of size c_j from the total number of cycle curves belonging to that woman.

Now, for each bootstrap sample it is necessary to calculate the eigenvalue decomposition of the resulting matrices TT^T and this will require about one and a half hours each. So, instead of using the non-parametric approach, they applied a partially parametric version of the bootstrap that required about 45 minutes to construct 35 bootstrap samples and their estimated fitted group means. In contrast to the Brumback an Rice approach, we decided to construct Bayesian confidence intervals. Recall that in Chapter II we showed that, by using the Bayesian approach, we are able to get $(1-\alpha)100\%$ Bayesian confidence intervals with the formula

$$\hat{f}_i(t) \quad \pm \quad z_{1-\alpha/2}\sqrt{\hat{\sigma}_e^2 a_{ii}}$$

where $\hat{\sigma}_e^2 = [(\boldsymbol{y}_i - \hat{\boldsymbol{f}}_i)^T(\boldsymbol{y}_i - \hat{\boldsymbol{f}}_i)]/(n-m)$, and the a_{ii} are the diagonal values of the corresponding matrix $(I - A_\lambda^i)$. Here, $\boldsymbol{y}_i$ denotes all the responses for group i and $\hat{\boldsymbol{f}}_i$ is the fitted function for the ith group mean function. The diagonal values can be calculated by using the $R_\epsilon(t)$ and the innovations produced by the algorithm (4.1.2). We will proceed to illustrate how to obtain them.

After applying the algorithm (4.1.2) to $\boldsymbol{y}_i$ and to each of the columns of the corresponding matrix T^i, we have the vectors $\hat{\boldsymbol{e}}_{\boldsymbol{y}_i}$, $\hat{\boldsymbol{e}}_{T_{i1}}$, $\hat{\boldsymbol{e}}_{T_{i2}}, \ldots, \hat{\boldsymbol{e}}_{T_{i(m-1)}}$, where T_{ik}, $k = 1, \ldots, (m-1)$ denotes the kth column of T^i. Arrange the vectors in a matrix like this

$$Z \quad = \quad \left[\begin{array}{cccc} \hat{\boldsymbol{e}}_{T_{i1}}, & \ldots, & \hat{\boldsymbol{e}}_{T_{i(m-1)}}, & \hat{\boldsymbol{e}}_{\boldsymbol{y}_i} \end{array} \right]. \tag{4.57}$$

Let $\boldsymbol{z}_i$, $i = 1, \ldots, n$, denote the ith row of Z. Then, the leverage values for time $t = i$ are given by

$$a_{ii} \quad = \quad R_\epsilon(i) - \boldsymbol{z}_i \boldsymbol{b}, \tag{4.58}$$

with $\boldsymbol{b}$ being the solution to the system of equations

$$Z^T T^i \boldsymbol{b} \;=\; Z^T[I : \hat{\boldsymbol{e}}\boldsymbol{y}_i].$$

Our 95% confidence bands help to give us a better idea of how the functions vary. The functions resulting from the permutation test in Figure 8 show some variability but it is hard to quantify it. Our estimators with respective confidence bands are shown in Figure 9.

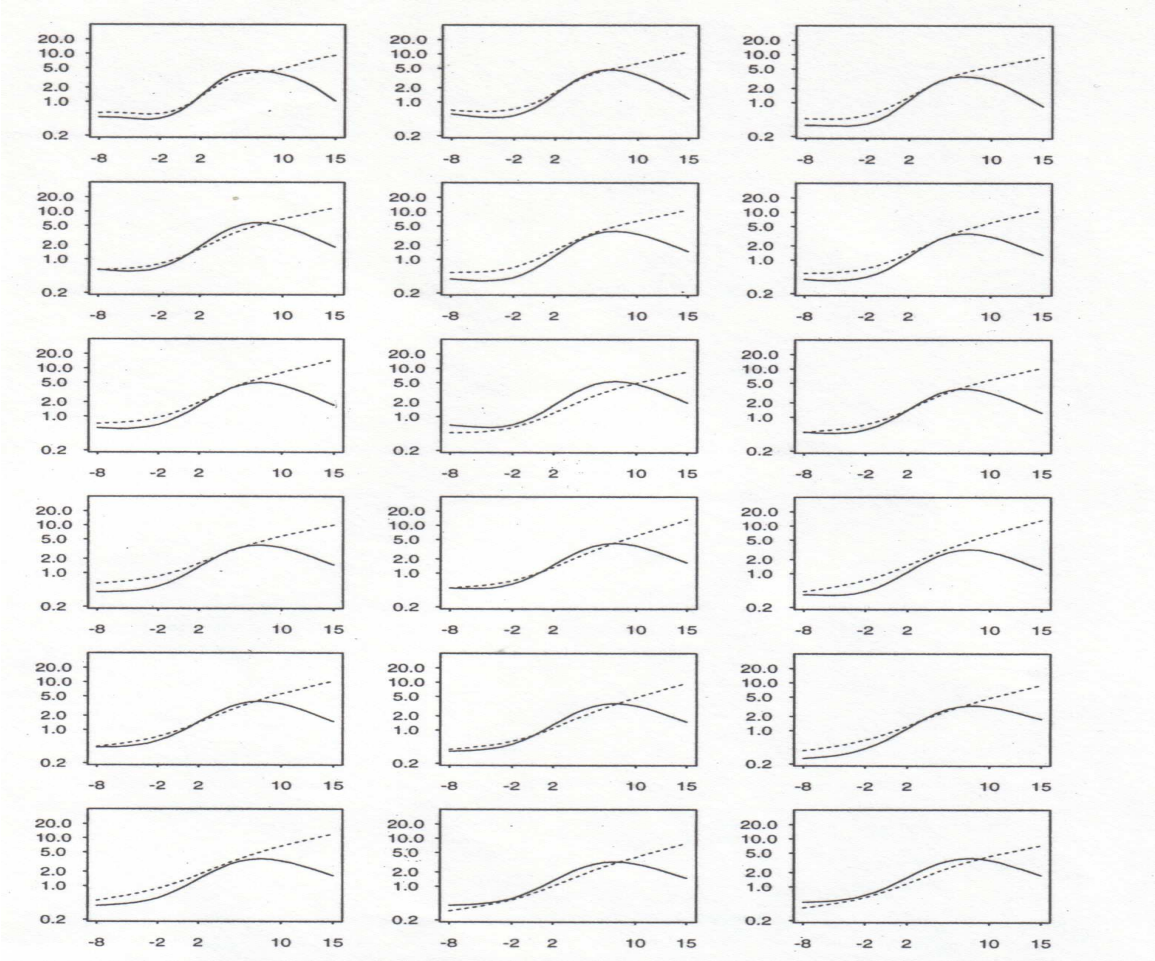

Figure 8: 35 bootstrap simulations to compare fitted group means. The original fit is displayed in the first panel for comparison. The picture was scanned from the Brumback and Rice article published by JASA (1998)

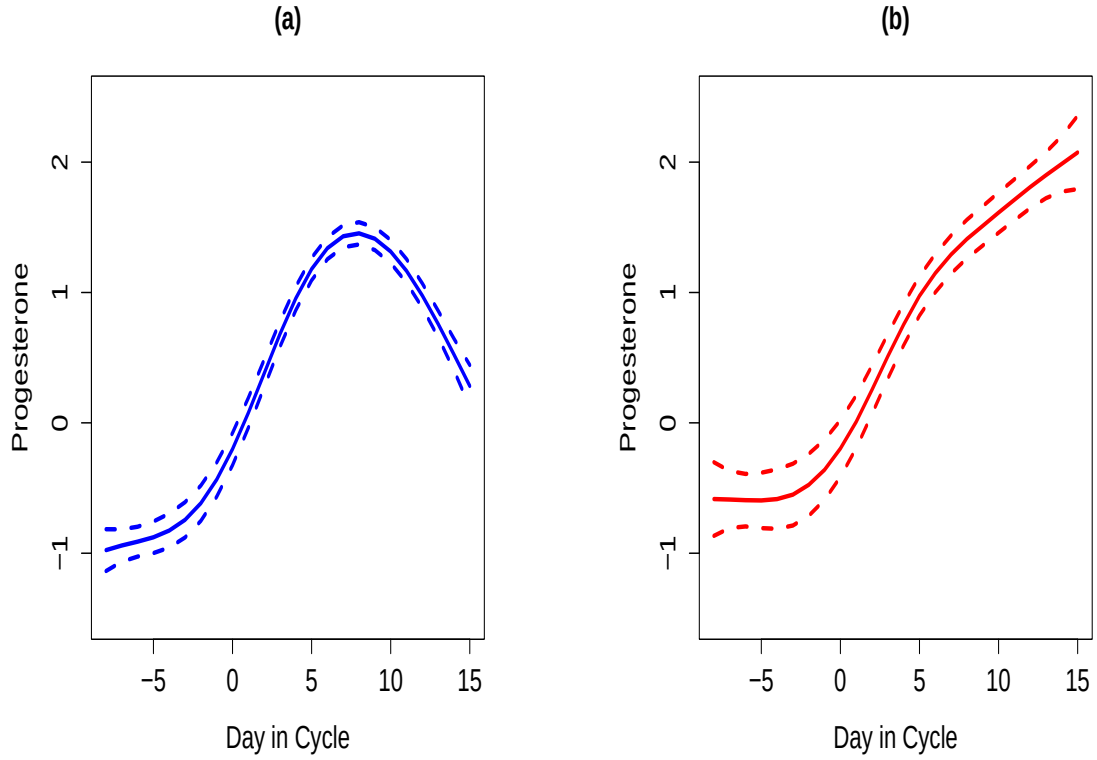

Figure 9: Smooth estimates for non conceptive, **(a)**, and conceptive, **(b)**, mean groups with respective 95% confidence bands.

Table 4 shows run time comparisons between the time it took to compute the fitted group mean function for the non conceptive group using the Kalman filter and the mixed-effects model with the SAS procedure Proc Mixed. The smoothing parameter was found via GML and the function optimize in R. The values of $\hat{\lambda} = 0.0005130205$ and $\hat{\sigma}_e^2 = .82$, were obtained from both, the Kalman filter and the mixed-effects model procedure. The calculation of the GML estimate of λ took about 1 minute CPU time in R.

Table 4: These are the run time comparisons between the Kalman filter and SAS PROC MIXED. I used the non conceptive group which has 1656 observations. The computations were done on a 2.00GHz processor with 512 MB of RAM memory.

Method	Real Time	CPU Time
Kalman Filter	1.14 secs.	0.9 secs.
PROC MIXED	29 mins.	27 mins.

.

4.3 Précis

In this chapter we have showed how to take advantage of the relationships established by theorem (3.1.1) between the PLS criterion and the Bayesian model to apply an efficient $O(n)$ Kalman filter algorithm. This algorithm is flexible enough to allow us to cover the case of correlated errors, when the errors can be represented in a state-space formulation. We also showed the computer time savings reached when applying the Kalman filter.

CHAPTER V

CONCLUSIONS AND FUTURE RESEARCH

5.1 Conclusions

When we started the research for this dissertation, we had very specific goals. Our intention was to provide a general framework to work specifically in the context of functional data analysis. We focused our attention on proposed functional models that were associated with popular inferential methods derived from the theory of mixed-effects models or smoothing splines. However, in establishing results for this case we realized that our approach could be generalized and applied in generic mixed-effects, penalized least squares or Bayesian model settings.

We began in Chapter II by showing the numerical equivalence of

a) the BLUP of $T\boldsymbol{\theta} + \boldsymbol{\gamma}$ from data following a mixed-effects model of the form

$$\boldsymbol{y} \;=\; T\boldsymbol{\theta} + \boldsymbol{\gamma} + \boldsymbol{e}, \tag{5.1}$$

with $\boldsymbol{y} = [y(t_1), \ldots, y(t_n)]^T$ the vector of responses, T the $n \times m$ matrix whose jth column is equal to $\boldsymbol{\phi_j} = [\phi_j(t_1), \ldots, , \phi_j(t_n)]^T$, for $j = 1, \ldots, m$, and $\phi_j(t_i) = \frac{t^{j-1}}{(j-1)!}$. The vector of fixed-effects $\boldsymbol{\theta}$ is an $m \times 1$ vector of unknown coeffincidents, $\boldsymbol{\gamma}$ is a random vector with zero mean and variance-covariance matrix $\sigma_b^2 R$, and independent of the random errors $\boldsymbol{e}$. The vector $\boldsymbol{e}$ is normally distributed with zero mean and variance-covariance matrix $\sigma_e^2 I$.

b) the smoothing spline estimator

$$\hat{\boldsymbol{f}} \;=\; \text{argmin}_{\boldsymbol{\theta},\boldsymbol{\gamma}} (\boldsymbol{y} - \boldsymbol{f})^T (\boldsymbol{y} - \boldsymbol{f}) + \lambda \boldsymbol{b}^T R \boldsymbol{b}, \tag{5.2}$$

for $\boldsymbol{f} = T\boldsymbol{\theta} + R\boldsymbol{\gamma}$, with the matrix T, $\boldsymbol{\theta}$, and $\boldsymbol{e}$ as in (5.1), $\boldsymbol{b}$ is a vector of unknown coeffincidents,

$$R = \left\{ \int_0^{\min(t_j,t_i)} \frac{(t_j-u)^{m-1}(t_i-u)^{m-1}}{[(m-1)!]^2} du. \right\}_{i,j=1,n}, \tag{5.3}$$

and λ the smoothing parameter.

c) the limit as $\nu \to \infty$ of the posterior mean of $T\boldsymbol{\theta} + \boldsymbol{\gamma}$ from the model

$$\boldsymbol{y} = T\boldsymbol{\theta} + \boldsymbol{\gamma} + \boldsymbol{e}, \tag{5.4}$$

with T, and $\boldsymbol{e}$ as before, and $\boldsymbol{\theta}$ is a random vector with a prior distribution given by $N(\mathbf{0}, \nu I)$. The random vector $\boldsymbol{\gamma} = \sigma_b \boldsymbol{X}$, with $X(t)$, $t \in [0,1]$, a zero-mean Gaussian stochastic process with covariance function

$$\mathrm{E}[X(t_i)X(t_j)] = \int_0^{\min(t_j,t_i)} \frac{(t_j-u)^{m-1}(t_i-u)^{m-1}}{[(m-1)!]^2} du.$$

The random vectors $\boldsymbol{\theta}$, $\boldsymbol{\gamma}$ and $\boldsymbol{e}$ are independent of each other.

The BLUP of (5.1), the smoothing spline estimator in (5.2), and the posterior mean of (5.4), all have the same form

$$[I - Q^{-1}(I - T(T^T Q^{-1} T)^{-1} T^T Q^{-1})]\boldsymbol{y}, \tag{5.5}$$

with $Q = n\lambda R + I$. These relationships have been know for some time now and they have been utilized by people like Wahba (1978, 1983), Wang (1996), and Brumback and Rice (1998) just to mention a few.

Accompanying these results, we have methods to estimate some of the parameters involved in (5.5): e.g., the smoothing parameter and the variance components. We have shown that the method of REML is equivalent to the GML method to estimate variance components. The method of GCV, UBR and GML are choices for estimating λ in (5.2).

In Chapter III, we generalized these results by considering a general mixed-effects model, the penalized least squares criterion and a general Bayesian model, i.e.,

a) the BLUP of $T\boldsymbol{\theta} + U\boldsymbol{\gamma}$ when

$$\boldsymbol{y} \;=\; T\boldsymbol{\theta} + U\boldsymbol{\gamma} + \boldsymbol{e}, \tag{5.6}$$

with T and U design matrices for the fixed and random effects, $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$, respectively, and the assumptions on the random vectors are as in (5.1),

b) the estimator $\hat{\boldsymbol{f}} = T\hat{\boldsymbol{\theta}} + U\hat{\boldsymbol{\gamma}}$ with $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\gamma}}$ obtained by minimizing

$$\mathrm{PLS}(\boldsymbol{\theta}, \boldsymbol{\gamma}) \;=\; (\boldsymbol{y} - T\boldsymbol{\theta} - U\boldsymbol{\gamma})^T(\boldsymbol{y} - T\boldsymbol{\theta} - U\boldsymbol{\gamma}) + \boldsymbol{\gamma}^T R_\lambda^{-1}\boldsymbol{\gamma}, \tag{5.7}$$

c) and the posterior mean of $T\boldsymbol{\theta} + U\boldsymbol{\gamma}$ from the Bayesian model

$$\boldsymbol{y} \;=\; T\boldsymbol{\theta} + U\boldsymbol{\gamma} + \boldsymbol{e}, \tag{5.8}$$

where T, U, $\boldsymbol{\gamma}$, and $\boldsymbol{e}$ are as in (5.6), the vector $\boldsymbol{\theta}$ is normally distributed with mean zero, variance-covariance matrix νW, for some positive-definite matrix W and ν is allowed to approach infinity. The vectors $\boldsymbol{\theta}$, $\boldsymbol{\gamma}$, $\boldsymbol{e}$ are independent of each other.

We have proved that (5.6), (5.7) and (5.8) yield the same numerical answer

$$[I - Q^{-1}(I - T(T^TQ^{-1}T)^{-1}T^TQ^{-1})]\boldsymbol{y}$$

with $Q = n\lambda URU^T + I$. This result allows us to compute any of the numerical solutions of (5.6), (5.7) or (5.8) using the other two settings. The choice of the estimation procedure to use then will depend entirely on computational efficiency or software availability.

Estimation of the variance components in the mixed-effects model setting or in the Bayesian model context, is usually done via GML. We showed that the GCV and UBR methods can also be utilized for estimation of the variance components. The

GML method is maximized by the ratio of the variance components, and the GCV and UBR approaches are minimized by the same ratio. A small Monte Carlo study was conducted to compare the GCV and GML techniques. The GML estimators showed a smaller root mean squared error (RMSE) than the GCV estimator in almost all cases. However, there were two cases when the GCV estimate of the random components had smaller RMSE than its GML counterpart.

In chapter IV we dealt with our second goal for this research: the implementation of an efficient algorithm to compute corresponding point and interval estimators. We introduced the notion of state-space models and showed that, if we can assume that our responses y have a state-space structure, then we can apply the Kalman filter algorithm.

The assumption on the y's is not as restrictive as it seems. When we assume a state-space structure on our responses, what we are doing is to assume a special structure in the variance-covariance matrix of our model, and this is what we usually do in practice, for example, assuming that the errors are generated by an autoregressive moving average (ARMA) process, Brownian motion, or just white noise. The cases mentioned above all have state-space representation.

We have illustrated the application of the Kalman fiter with two cases: one in which the errors are generated by an AR(1) process, and the other a functional linear model. We have provided the code for these cases in Appendix A and B. We also provided run time comparisons between the estimators obtained via Kalman filter and by using the SAS procedure Proc Mixed. Our results show the great savings in computer time when using the Kalman filter.

5.2 Future Research

We have seen the great advantage of applying the Kalman filter when computing BLUP's in the mixed-effects model context or smoothing spline estimators. Due to its computational efficiency, application of the Kalman filter in state-space oriented problems has increased. There is an active area of research in areas where, like atmospheric or oceanic sciences, besides the temporal dimension there exists the influence of a spatial effect (see, e.g., Wickle and Cressie, 1999; Huang et al., 2002). A known approach for this type of spatial-temporal problems is the use multi-resolution analysis (MRA). It would be of interest to explore the random-effects model and penalized least squares regression in the MRA setting and study its connection to kriging.

We would like to generalized the techniques used in this dissertation to the varying coefficient models. Eubank et al. (2004) considered the use of smoothing splines as estimators of coefficient curves for varying coefficient models with a single effect modifying covariate. They showed that smoothing spline estimators for varying coefficient models can be represented using the Bayesian setting discussed here and hence, the Kalman filter can be applied to obtain the estimators. It would be of interest to apply the methods found in this dissertation to that case and to extend it to the case with more than one modifying covariate.

In this dissertation we briefly explored the sampling distribution of the GCV estimate. We would like to explore in more detail the sampling and asymptotic properties of the GCV and UBR estimates of the variance components in the mixed-effects models.

REFERENCES

Anselone, P. and Laurent, P. (1968). A general method for the construction of interpolating or smoothing spline functions. *Numerische Mathematik* **12**, 66–82.

Ansley, C. and Kohn, R. (1985). Estimation, filtering, and smoothing in state space models with incompletely specified initial conditions. *The Annals of Mathematical Statistics* **13**, 1286–1316.

Box, G. and Jenkins, G. (1976). *Time Series Analysis*. San Francisco: Holden–Day.

Brumback, B. and Rice, J. (1998). Smoothing spline models for the analysis of nested and crossed samples of curves. *Journal of the American Statistical Association* **93**, 961–993.

Corbeil, R. and Searle, S. (1976). Restricted maximum likelihood (REML) estimation of variance components in the mixed models. *Technometric* **18**, 31–38.

Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik* **31**, 377–403.

De Jong, P. (1988). The likelihood for a state space model. *Biometrika* **75**, 165–169.

De Jong, P. (1989). Smoothing and interpolation with the state-space model. *Journal of the American Statistical Association* **84**, 1085–1088.

Demidenko, E. (2004). *Mixed Models, Theory and Application*. Hoboken, New Jersey: Wiley Inc. & Sons.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B* **39**, 1–22.

Eisenhart, C. (1947). The assumptions underlying the analysis of variance. *Biometrics* **3**, 1–21.

Eubank, R. (1988). *Spline Smoothing and Non Parametric Regression* 1st edition. New York: Marcel Dekker Inc.

Eubank, R. (1996). *Nonparametric Regression and Spline Smoothing* 2nd edition. New York: Marcel Dekker Inc.

Eubank, R., Huang, C., Muñoz Maldonado, Y., Wang, N., Wang, S., and Buchanan, R. (2004). Smoothing spline estimation in varying coefficient models. *Journal of the Royal Statistical Society, Series B* **66**, 653–667.

Eubank, R., Huang, C., and Wang, S. (2003). Adaptive order selection for spline smoothing. *Journal of Computational and Graphical Statistics* **12**, 382–397.

Eubank, R. and S., W. (2002). The equivalence between the Cholesky decomposition and the Kalman filter. *The American Statistician* **56**, 39–43.

Gantmakher, F. R. (1959). *The Theory of Matrices*. New York: Chelsea Pub. Co.

Graybill, F. (1976). *The Theory and Application of the Linear Model*. North Scituate, Massachusetts: Duxbury.

Greville, T. (1969). Introduction to spline functions. In T. Greville (ed.), *Theory and Applications of Spline Functions*. New York: Academic Press.

Guo, W. (2002). Functional mixed effects models. *Biometrics* **58**, 121–128.

Guo, W. (2003). Functional data analysis in longitudinal settings using smoothing splines. *Statistical Methods in Medical Research* **13**, 1–24.

Hartley, H. and Rao, J. (1967). Maximum likelihood estimation for the mixed analysis of variance models. *Biometrics* **54**, 93–108.

Harville, D. (1976). Extension of the Gauss-Markov theorem to include the estimation of random effects. *The Annals of Mathematical Statistics* **4**, 384–395.

Harville, D. (1977). Maximum likelihood approach to variance component estimation and to related problems. *Journal of the American Statistical Association* **72**, 320–338.

Heckman, N. (1997). The theory and applications of penalized least squares methods or reproducing kernel Hilbert spaces made easy..

Henderson, C. (1953). Estimation of variance and variance components. *Biometrics* **9**, 226–252.

Henderson, C., Kempthorne, O., Searle, S., and Krosigk, C. (1959). The estimation of environmental and genetic trends from records subject to culling. *Biometrics* **15**, 192–218.

Hocking, R. (1996). *Methods and Applications of Linear Models, Regression and the Analysis of Variance*. New York: Wiley.

Householder, A. (1964). *The Theory of Matrices in Numerical Analysis*. New York: Dover.

Huang, H., Johannesson, G., and Cressie, N. (2002). Fast, resolution-consistent spatial prediction of global processes from satellite data. *Journal of Computational and Graphical Statistics* **11**, 63–88.

Khon, R. and Ansley, C. (1987). A new algorithm for spline smoothing based on smoothing a stochastic process. *SIAM Journal of Scientific and Statistical Computing* **8**, 33–48.

Khon, R., Ansley, C., and Tharm, D. (1991). The performance of cross-validation and maximum likelihood estimators of spline smoothing parameters. *Journal of the American Statistical Association* **86**, 1042–1050.

Kimeldorf, G. and Wahba, G. (1970). A correspondence between bayesian estimation on stochastic process and smoothing by splines. *The Annals of Mathematical Statistics* **2**, 495–502.

Kohn, R. and Ansley, C. (1989). A fast algorithm for signal extraction, influence and cross-validation in state-space models. *Biometrika* **76**, 65–79.

Koopman, S. and Durbin, J. (1998). Fast filtering and smoothing for multivariate state space models. *Journal of Times Series Analysis* **21**, 281–296.

Patterson, H. and Thompson, R. (1971). Recovery of interblock information when cell sizes are unequal. *Biometrika* **58**, 545–554.

Reinsch, C. (1967). Smoothing by spline functions. *Numerische Mathematik* **10**, 177–183.

Robinson, G. (1991). That BLUP is a good thing: The estimation of random effects. *Statistical Science* **6**, 15–51.

Sallas, W. and Harville, D. (1981). Best linear recursive estimation for mixed linear models. *Journal of the American Statistical Association* **76**, 860–869.

Scheffé, H. (1959). *The Analysis of Variance* (Wiley classics library ed. edition). New York: Wiley-Interscience Publication.

Schoenberg, I. (1964). On interpolation by spline functions and its minimum properties. *Numerical Analysis* **5**, 109–129.

Schumaker, L. (1981). *Spline Functions*. New York: Wiley.

Silverman, B. (1985). Some aspects of the spline smoothing approach to nonparametric regression curve fitting (with discussion). *Journal of the Royal Statistical Society, Series B* **47**, 1–52.

Speckman, P. (1985). Spline smoothing and optimal rates of convergence in nonparametric regression models. *The Annals of Mathematical Statistics* **13**, 970–983.

Speed, T. (1991). That BLUP is a good thing: The estimation of random effects: Comment. *Statistical Science* **6**, 42–44.

Tukey, J. (1956). Variances of variance components: I. Balanced designs.. *The Annals of Mathematical Statistics* **27**, 722–736.

Wahba, G. (1978). Improper priors, spline smoothing, and the problem of guarding against model errors in regression. *Journal of the Royal Statistical Society, Series B* **40**, 364–372.

Wahba, G. (1983). Bayesian confidence intervals for the cross-validated smoothing spline. *Journal of the Royal Statistical Society, Series B* **45**, 133–150.

Wahba, G. (1985). A comparison of GCV and GML for choosing the smoothing parameter in the generalized spline smoothing problem. *The Annals of Mathematical Statistics* **13**, 1378–1402.

Wahba, G. (1990). *Spline Models for Observational Data*, Vol. 59. Philadelphia, Pennsylvania: SIAM.

Wang, Y. (1996). *Smoothing spline models with correlated random errors.* Technical Report 996, Department of Statistics, University of Wisconsin-Madison, Madison, Wisconsin.

Wang, Y. (1998a). Mixed effects smoothing spline analysis of variance. *Journal of the Royal Statistical Society, Series B* **60**, 159–174.

Wang, Y. (1998b). Smoothing spline models with correlated random errors. *Journal of the American Statistical Association* **93**, 341–348.

Wickle, C. and Cressie, N. (1999). A dimension-reduced approach to space-time Kalman filtering. *Biometrika* **86**, 815–829.

APPENDIX A

SAS PROGRAMS

fmat

```
/*************************************************************************/
/*  begin module fmat to compute the transition matrix F in the KF.  */
/*  it has as arguments the vector of ordinates ti, the order of the */
/*  derivative in the penalty term or the number of fixed parameters */
/*  to estimate in the mixed models effects and the number of        */
/*  observations at which F(t) is computed.                          */
/*  It requires to initialize the proc iml software.                 */
/*************************************************************************/

*call to iml;

proc iml;

*This makes the storage library for modules and matrices;
*in the Kalman library;
*and the storage kstor;

reset storage=kalman.kstor;

*This initializes the module;
start fmat(t,m,k);
    /*TODO: check that the values are valid*/

    nr=nrow(t);
    nc=ncol(t);
    mr=round(m,1);

    *checks for k being a positive integer since it ;
    *indicates the kth point to evaluate in F;
    if  k<1 then  do;
        print "k must be positive integers";
        stop;
    end;
    *check that the order of derivative is a positive integer;
    if m<1 | m^=mr then do;
```

```
        print m "m  must be positive integers";
        stop;
    end;
    *Check that t is an n by 1 vector;
    if nc^=1 then do;
        print "t must be an n by 1 vector";
        stop;
    end;
    *check that the kth point is inside the length of t;
    if k>nr then do;
        print "k must be a number between 1 and the length of t";
        stop;
    end;


    /* get values for t_k and t_k-1; */
    t_k = t[k,1];
    if ( k = 1 ) then t_k1 = t_k;
    else t_k1 = t[k-1,1];

    *print t_k;
    *print t_k1;

    f=I(m);
    *print f;
    *start the loop to make the F matrix;
    do r=1 to m;
        do c=(r+1) to m;
            a=c-r;
            b=gamma(a+1);
            d=t_k - t_k1;
            f[r,c]=(d**a)/b;
        end;
    end;

return (f);
finish fmat;

*This stores the module fmat in kalman.kstor and shows that;
* it is in there;

store module=(fmat);
```

```
*show storage;
quit;
```

umat

```
/*************************************************************************/
/*  begin module umat to compute the covariance matrix U in the KF.   */
/*  it has as arguments the vector of ordinates ti, the order of the  */
/*  derivative in the penalty term or the number of fixed parameters  */
/*  to estimate in the mixed models effects and the number of         */
/*  observation at which F(t) is computed and the smoothing parameter */
/*   lambda. It requires to initialize the proc iml software.         */
/*************************************************************************/

proc iml;


*This makes the storage library for modules and matrices;
* in the Kalman library and the storage kstor;
reset storage=kalman.kstor;

*This initializes the module;
start umat(t,m,k,lambda);

    /*TODO: check that the values are valid*/
    mr=round(m,1);
    nr=nrow(t);
    nc=ncol(t);


    *checks for k being a positive integer since it indicates the kth;
    *point to evaluate in F;
    if  k<1 then  do;
        print k " k must be positive integers";
        stop;
    end;

    *check that the order of derivative is a positive integer;
    if m<1 | m^=mr then do;
        print m "m  must be positive integers";
        stop;
    end;
```

```
*Check that t is an n by 1 vector;
if nc^=1 then do;
    print "t must be an n by 1 vector";
    stop;
end;

*check that the kth point is inside the length of t;
if k>nr then do;
    print  k "k must be a number between 1 and the length of t: " nr;
    stop;
end;

*check that lambda is bigger than 0;
if lambda<0 then do;
    print lambda "lambda must be non negative";
    stop;
end;


/* get values for t_k and t_k-1; */
t_k = t[k,1];
if ( k = 1 ) then t_k1 = t_k;
else t_k1 = t[k-1,1];

*print t_k;
*print t_k1;

u=j(m,m,0);

*start the loop to make the F matrix;
do r=1 to m;
    do c=1 to m;
        ex1=2*m-c-r+1;
        fac1=fact(m-r);
        fac2=fact(m-c);
        d=t_k - t_k1;
        u[r,c]=(d**ex1)/(ex1*fac1*fac2);
    end;
end;

u=lambda*u;
```

```
    return (u);

finish umat;

*This stores the module umat in kalman.kstor;
* and shows that it is in there;

store module=(umat);
*show storage;
quit;
```

polymat

```
/*************************************************************************/
/*  This function creates a matrix of polynomial basis #t_i^j/j! for */
/*  i=1,...n and j=0,...,m-1.                                            */
/*  t  is a vector of times or distances t and                           */
/*  m is the order of the polynomial                                     */
/*************************************************************************/

proc iml;

*This makes the storage library for modules and matrices;
*in the Kalman library and the storage kstor;
reset storage=kalman.kstor;

*This initializes the module;
start polymat(t,m);

    *checking that all arguments are correct;

    nr=nrow(t);
    nc=ncol(t);
    mr=round(m,1);



   *check that the order of derivative is a positive integer;
    if m<1 | m^=mr then do;
        print "m  must be a positive integer " m;
        stop;
    end;
```

```
    *Check that t is an n by 1 vector;
    if nc^=1 then do;
        print "t must be an n by 1 vector";
        stop;
    end;

    PB=j(nr,m,.);
    PB[,1]=1;
    do i=2 to m;
    expo=i-1;
        PB[,i]=t##expo/fact(expo);
    end;
    return(PB);

finish polymat;

*This stores the module polymat in kalman.kstor;
* and shows that it is in there;

store module=(polymat);
*show storage;
quit;
```

kff

```
/*************************************************************************/
/*  This function computes the forward recursion of the Kalman filter */
/*  to obtain the estimator of a function. ti is the vector of        */
/*  ordinates, y is the vector of responses, m is the mth             */
/*  derivative in the penalty term, lambda is the smoothing parameter */
/*  and w is the weight for the random errors.                        */
/*  This is a subroutine so we need to pass the names of the matrices */
/*  we want to obtain which : the innovations=inov, the temporary     */
/*  matrix, K=kmat, which is necessary for the backward recursion,    */
/*  res=r the residuals, and the vector stvec to check the program.   */
/*************************************************************************/



proc iml;

*This makes the storage library for modules and matrices;
* in the Kalman library and the storage kstor;
```

```
reset storage=kalman.kstor;

*This initializes the module;
start kff(inov,kmat,res,stvec,t,y,m,lambda,w,dev);

    ni=nrow(t);
    Ti=polymat(t,m);
    Z=j(ni,(m+1),0);
    X=j(m,(m+1),0);
    S=j(m,m,0);
    K=j(m,ni,0);
    a=j(ni,1,0);
    r=j(ni,1,0);
    h=j(m,1,0);
    h[dev,1]=1;
    Z[,1:m]=Ti;
    Z[,m+1]=y;
    W=j(ni,1,w);
        do i=1 to ni;
            F=Fmat(t,m,i);
            U=Umat(t,m,i,lambda);
            Ft=F‘;
            ht=h‘;
            S=(F*S*Ft)+U;
            r[i,1]=(ht*S*h)+W[i,1];
            Z[i,]=Z[i,]-(ht*F*X);
            X=F*X+(S*h*Z[i,]/r[i,]);
            a[i,1]=X[1,m+1];
            K[,i]=F*S*h/r[i,1];
            S=S-S*h*ht*S/r[i,1];
        end;

    inov=Z;
    res=r;
    kmat=K;
    stvec=a;

finish kff;

*This stores the module kff in kalman.kstor;
* and shows that it is in there;
```

```
store module=(kff);
*show storage;
quit;
```

kfb

```
/*********************************************************************/
/*  This function computes the backward recursion of the Kalman filter*/
/*  to obtain the smooth estimator of a function. ti is the vector of */
/*  ordinates, y is the vector of responses, m is the mth derivative  */
/*  in the penalty  term, lambda is the smoothing parameter and w is  */
/*  the weight for the random errors.                                 */
/*  inov is the matrix of innovations from the forward recursion, kmat*/
/*  is the matrix K necessary to compute the variances s(t|t), and the*/
/*  residuals res.                                                    */
/*  The program will return the modified matrix inov, and r.          */
/*********************************************************************/



proc iml;

*This makes the storage library for modules and matrices;
* in the Kalman library and the storage kstor;
reset storage=kalman.kstor;

*This initializes the module;
start kfb(inovs,ress,inov,kmat,res,t,m,dev);

    n=nrow(t);
    Z=inov;
    K=kmat;
    r=res;
    X=j(m,(m+1),0);
    S=j(m,m,0);
    b=Z[n,];
    c=r[n,1];
    r[n,1]=1-(1/r[n,1]);
    Z[n,]=Z[n,]/c;
    h=j(m,1,0);
    h[dev,1]=1;
    do i=(n-1) to 1 by -1;
        F=Fmat(t,m,(i+1));
        X=(-(h*b)/c)+(F-K[,i+1]*h`)`*X;
        S=(h*h`/c)+(F-K[,i+1]*h`)`*S*(F-K[,i+1]*h`);
        b=Z[i,];
        c=r[i,1];
```

```
        Z[i,]=(Z[i,]/c)+(X`*K[,i])`;
        r[i,1]=1-(1/r[i,1])-K[,i]`*S*K[,i];
     end;

     inovs=Z;
     ress=r;

finish kfb;


*This stores the module kff in kalman.kstor;
* and shows that it is in there;

store module=(kfb);
*show storage;
quit;
```

kf

```
/*************************************************************************/
/*  This code calls the kalman filter forward and backward recursions*/
/*  and obtains the fitted values and their respective residuals of a*/
/*  smooth function f.                                                   */
/*  t is the vector of ordinates.                                        */
/*  y is the vector of responses.                                        */
/*  m is the order of the derivative in the penalty term.                */
/*  lambda is the smoothing parameter.                                   */
/*  dev is the order of the derivative we want to estimate,.i.e,         */
/*  if dev=1 then we are estimating f, if dev=2 we are estimating f' */
/*  , etc. and w is the weights for the random errors.                   */
/*************************************************************************/


proc iml;

*This makes the storage library for modules and matrices;
* in the Kalman library and the storage kstor;
reset storage=kalman.kstor;

*This initializes the module;
start kf(fhat,r,inovs,inov,kmat,ress,res,stvec,
         t,y,m,lambda,w,dev,update);
```

```
load module=(kff kfb fmat umat polymat);

run kff(inov,kmat,res,stvec,t,y,m,lambda,dev,w);
run kfb(inovs,ress,inov,kmat,res,t,m,dev);

n=nrow(t);
*This gives Q^(-1)y;
Z=inovs[,m+1];
*this gives Q^(-1)Ti;
V=inovs[,1:m];
Ti=polymat(t,m);
C=j(n,n+1,1);
iden=i(n);
C[,1:n]=iden;
C[,(n+1)]=Z;
a=Ti`*V;
b=Ti`*C;
*This solves the system Ti^TV=Ti^T[I:z];
B=solve(a,b);
fhat=Z-V*B[,(n+1)];
fhat=y-fhat;
*This computes the residual values;
update=inovs[,1:m]*B[,1:n];
temp1=j(n,1,1);
temp2=diag(update)*temp1;
r=ress+temp2;

finish kf;
```

APPENDIX B

R PROGRAMS

Fmat

```
Fmat=function(ti,m,j){
    #this function computes the transition matrix in the state-space
    # model
    #ti is the vector of time or distance points
    #m is the number of derivatives in the penalty term
    #j is the point at which we are obtaining the matrix.

    ni=length(ti)
    if (!is.numeric(ti)) {
        stop("ti needs to be a numeric vector")
        }
    else if (!m>0) {
        stop("m must be a nonnegative integer")
        }

    else if (j==0 || j>ni) {
        stop("the range of j must be an integer between 1 and length
        of ti")
        }
    #initial value t0 equals the first value of input vector ti.
    t0=ti[1]
    #including the value t0 in ti
    ti=c(t0,ti)
    #reindexing the jth point of ti
    j=j+1
    Fi=matrix(0,m,m)
    for(r in 1:m){
        for(c in r:m){
            Fi[r,c]=((ti[j]-ti[j-1])^(c-r))/factorial(c-r)
        }
    }
    return(Fi)
}
```

Umat

```
Umat=function(ti,m,j,lambda){
```

```
    #this function computes the covariance matrix of the
    #wiener process u(t_i).
    #ti is the vector of time or distance points
    #m is the number of derivatives in the penalty term
    #j is the point at which we are obtaining the matrix.

    ni=length(ti)
    if (!is.numeric(ti)) {
        stop("ti needs to be a numeric vector")
        }
    else if (!m>0) {
        stop("m must be a nonnegative integer")
        }

    else if (j==0 || j>ni) {
        stop("the range of j must be an integer between 1 and length of
         ti")
        }

    else if (lambda<0 || lambda==0) {stop("lambda must be nonnegative")}
    Ui=matrix(NA,m,m)
    #initial value t0 equals the first value of input vector ti.
    t0=ti[1]
    #including the value t0 in ti
    ti=c(t0,ti)
    #reindexing the jth point of ti
    j=j+1
    for(r in 1:m){
        for(c in 1:m){
            Ui[r,c]=(ti[j]-ti[j-1])^(2*m-r-c+1)/((2*m-r-c+1)
            *factorial(m-r)*factorial(m-c))
            }
        }
        Ui=(lambda)*Ui
    return(Ui)
}
```

kff

```
kff=function(ti,y,m,lambda,w){

    #This function computes the forward recursion
    #of the Kalman filter to obtain the estimator of a function.
```

```
    #ti is the vector of ordinates.
    #y is the vector of responses.
    #m is the mth derivative in the penalty term.
    #lambda is the smoothing parameter.
    #w is the weight for the random errors.

    ni=length(ti)
    Ti=polymat(ti,m)
    Z=matrix(NA,ni,(m+1))
    X=matrix(0,m,(m+1))
    S=matrix(0,m,m)
    K=matrix(NA,m,ni)
    a=matrix(NA,ni,1)
    r=rep(0,ni)
    h=rep(0,m)
    h[1]=1
    Z[,1:m]=Ti
    Z[,m+1]=y
    W=rep(w,ni)
    for(i in 1:ni){
        F=Fmat(ti,m,i)
        U=Umat(ti,m,i,lambda)
        S=F%*%S%*%t(F)+U
        r[i]=t(h)%*%S%*%h+W[i]
        Z[i,]=Z[i,]-t(h)%*%F%*%X
        X=F%*%X+(S%*%h%*%Z[i,]/r[i])
        a[i,1]=X[1,m+1]
        K[,i]=F%*%S%*%h/r[i]
        S=S-S%*%h%*%t(h)%*%S/r[i]
        }
    #added X and S in the return.
    list(Z=Z,r=r,K=K,a=a)
}
```

kfb

```
kfb=function(ti,m,Z,r,K){
    #This function computes the backward recursion
    #of the Kalman filter to obtain the estimator of a function.
    #ti is the vector of ordinates.
    #m is the mth derivative in the penalty term.
    #Z is the matrix of innovations at time t_i|t_(i-1),...,t_1
```

```
    #r is the vector of variances for the innovations
    #K is the matrix of updates variances of the state vector.

    n=length(ti)
    X=matrix(0,m,(m+1))
    S=matrix(0,m,m)
    b=Z[n,]
    c=r[n]
    r[n]=1-(1/r[n])
    Z[n,]=Z[n,]/c
    a=matrix(NA,n,1)
    h=rep(0,m)
    dev=1
    h[dev]=1
    for(i in (n-1):1)
        {
        F=Fmat(ti,m,(i+1))
        X=(-(h%*%t(b))/c)+t(F-K[,i+1]%*%t(h))%*%X
        S=(h%*%t(h)/c)+t(F-K[,i+1]%*%t(h))%*%S%*%(F-K[,i+1]%*%t(h))
        b=Z[i,]
        c=r[i]
        Z[i,]=(Z[i,]/c)+t(X)%*%K[,i]
        r[i]=1-(1/r[i])-t(K[,i])%*%S%*%K[,i]
        a[i,1]=X[1,m+1]
        }

    list(Z=Z,r=r,a=a)

}
```

kf

```
kf=function(ti,y,m,lambda,w){
    #This code estimates a single smooth function
    #via Kalman Filtering.
    #ti is the vector of ordinates
    #y is the vector of responses
    #m is the mTh derivative in the penalty term
    #lambda is the smoothing parameter
    #w is the weight of random errors.
```

```
    ni=length(ti)
    #lambdahat=gml(
    #this computes the forward recursion of the Kalman filter.
    xf=kff(ti,y,m,lambda,w)
    #this computes the backward recursion of the Kalman filter.
    xb=kfb(ti,m,xf$Z,xf$r,xf$K)
    #this gives Q^(-1)y
    z=xb$Z[,(m+1)]
    #this gives Q^(-1)Ti
    V=xb$Z[,1:m]
    V=as.matrix(V)
    Ti=polymat(ti,m)
    C=matrix(NA,ni,ni+1)
    iden=diag(1,ni)
    C[,1:ni]=iden
    C[,(ni+1)]=z
    a=t(Ti)%*%V
    b=t(Ti)%*%C
    #This solves the system Ti^TV=Ti^T[I:z]
    B=solve(a,b)
    fhat=z-V%*%B[,(ni+1)]
    fhat=y-fhat
    #This computes the residual values
    r=xb$r
        #for(i in 1:ni)
        #{
        #r[i]=r[i]+xb$Z[i,1:m]%*%B[,i]
        #}
    update=diag(xb$Z[,1:m]%*%B[,1:ni])
    r=r+update
    list(fhat=fhat,r=r,lambda=lambda,X=xf$a,Xb=xb$a,Zf=xf$Z,Zb=xb$Z)

}
```

gmll

```
gmll=function(lam,data)
    {
    #this function maximizes the likelihood with respect to lambda
    x=kff(ti=data$ti,y=data$y,m=data$m,lambda=lam,w=data$w)
    R=sum(log(x$r))
    Rinv=diag(1/x$r)
    a=t(x$Z[,data$m+1])%*%Rinv%*%x$Z[,data$m+1]
```

```
    lik=-2*(R+a)
    return(lik)
    }
```

The following are the modified Kalman recursions for the case of autocorrelate of order 1.

kff

```
kff=function(ti,y,m,lambda,w,q){

    #This function computes the forward recursion
    #of the Kalman filter to obtain the estimator of a function.
    #ti is the vector of ordinates.
    #y is the vector of responses.
    #m is the mth derivative in the penalty term.
    #lambda is the smoothing parameter.
    #w is the weight for the random errors.
    #q is the correlation for the errors.

    ni=length(ti)                #number of observations
    Ti=polymat(ti,m)             #matrix T of polynomial terms
    Z=matrix(NA,ni,(m+1))        #innovations
    X=matrix(0,(m+1),(m+1))      #state-vector
    S=matrix(0,(m+1),(m+1))      #variance of the state-vectors
    K=matrix(NA,(m+1),ni)        #Kalman gain
    a=matrix(NA,ni,2)            #storage matrix for the state-vectors
    r=rep(0,ni)                  #variance of the innovations
    h=rep(0,(m+1))               #vector in the observation equation
    h[1]=1
    h[m+1]=1
    Z[,1:m]=Ti
    Z[,m+1]=y
    W=rep(w,ni)               #vector of variances for the iid errors.
    F=matrix(0,(m+1),(m+1))
    U=matrix(0,(m+1),(m+1))
    for(i in 1:ni){
        Fx=Fmat(ti,m,i)
        F[1:m,1:m]=Fx
        F[(m+1),(m+1)]=q
        Ux=Umat(ti,m,i,lambda)
        U[1:m,1:m]=Ux
```

```
        U[(m+1),(m+1)]=W[i]
        S=F%*%S%*%t(F)+U
        r[i]=t(h)%*%S%*%h+1
        Z[i,]=Z[i,]-t(h)%*%F%*%X
        X=F%*%X+(S%*%h%*%Z[i,]/r[i])
        a[i,1]=X[1,m+1]
        a[i,2]=X[1,1]
        K[,i]=F%*%S%*%h/r[i]
        S=S-S%*%h%*%t(h)%*%S/r[i]
        }
    #added X and S in the return.
    list(Z=Z,r=r,K=K,a=a)
}
```

kfb

```
kfb=function(ti,m,Z,r,K,q){
    #This function computes the backward recursion
    #of the Kalman filter to obtain the estimator of a function.
    #ti is the vector of ordinates.
    #m is the mth derivative in the penalty term.
    #Z is the matrix of innovations at time t_i|t_(i-1),...,t_1
    #r is the vector of variances for the innovations
    #K is the matrix of updates variances of the state vector.

    n=length(ti)
    X=matrix(0,(m+1),(m+1))
    S=matrix(0,(m+1),(m+1))
    b=Z[n,]
    c=r[n]
    r[n]=1-(1/r[n])
    Z[n,]=Z[n,]/c
    a=matrix(NA,n,2)
    a[n,1]=X[1,m+1]
    a[n,2]=X[1,1]
    h=rep(0,m+1)
    dev=1
    h[dev]=1
    h[m+1]=1
    F=matrix(0,(m+1),(m+1))
    for(i in (n-1):1)
        {
        Fx=Fmat(ti,m,(i+1))
```

```
        F[1:m,1:m]=Fx
        F[(m+1),(m+1)]=q
        X=(-(h%*%t(b))/c)+t(F-K[,i+1]%*%t(h))%*%X
        S=(h%*%t(h)/c)+t(F-K[,i+1]%*%t(h))%*%S%*%(F-K[,i+1]%*%t(h))
        b=Z[i,]
        c=r[i]
        Z[i,]=(Z[i,]/c)+t(X)%*%K[,i]
        r[i]=1-(1/r[i])-t(K[,i])%*%S%*%K[,i]
        a[i,1]=X[1,m+1]
        a[i,2]=X[1,1]
        }

    list(Z=Z,r=r,a=a)

}
```

kf

```
kf=function(ti,y,m,lambda,w,q){
    #This code estimates a single smooth function
    #via Kalman Filtering.
    #ti is the vector of ordinates
    #y is the vector of responses
    #m is the mTh derivative in the penalty term
    #lambda is the smoothing parameter
    #w is the weight of random errors.


    ni=length(ti)
    #lambdahat=gml(
    #this computes the forward recursion of the Kalman filter.
    xf=kff(ti,y,m,lambda,w,q)
    #this computes the backward recursion of the Kalman filter.
    xb=kfb(ti,m,xf$Z,xf$r,xf$K,q)
    #this gives Q^(-1)y
    z=xb$Z[,(m+1)]
    #this gives Q^(-1)Ti
    V=xb$Z[,1:m]
    V=as.matrix(V)
    Ti=polymat(ti,m)
    C=matrix(NA,ni,ni+1)
    iden=diag(1,ni)
    C[,1:ni]=iden
```

```
    C[,(ni+1)]=z
    a=t(Ti)%*%V
    b=t(Ti)%*%C
    #This solves the system Ti^TV=Ti^T[I:z]
    B=solve(a,b)
    fhat=z-V%*%B[,(ni+1)]
    fhat=y-fhat
    #This computes the residual values
    r=xb$r
        #for(i in 1:ni)
        #{
        #r[i]=r[i]+xb$Z[i,1:m]%*%B[,i]
        #}
    update=diag(xb$Z[,1:m]%*%B[,1:ni])
    r=r+update
    list(fhat=fhat,r=r,lambda=lambda,X=xf$a,Xb=xb$a,Zf=xf$Z,Zb=xb$Z)

}
```

VITA

Yolanda Muñoz Maldonado, was born in Acapulco, Guerrero, Mexico, on June 19, 1970. She received a B.S. degree in 1997 from the Universidad Autonoma de Yucatan in Mathematical Education and a M.S. degree in Statistics in 2000 from The University of Texas at El Paso. In 2000, she enrolled in Texas A&M University to pursue a Ph.D. in Statistics. Her research interests include Functional Data Analysis, Mixed-effects Models and Kalman Filtering.

Permanent address:

Yolanda Muñoz Maldonado
C. 16 No. 154 esq. con 21,
Merida, Yucatan, Mexico, CP 79249
E-mail: Y_ Munoz70@hotmail.com.

This dissertation was typed by Yolanda Muñoz Maldonado.